

NASA Technical Memorandum 103719

IN-71
163199
p-161

In-Flight Near- and Far-Field Acoustic Data Measured on the Propfan Test Assessment (PTA) Testbed and With an Adjacent Aircraft

Richard P. Woodward and Irvin J. Loeffler
Lewis Research Center
Cleveland, Ohio

April 1993

(NASA-TM-103719) IN-FLIGHT NEAR-
AND FAR-FIELD ACOUSTIC DATA
MEASURED ON THE PROPFAN TEST
ASSESSMENT (PTA) TESTBED AND WITH
AN ADJACENT AIRCRAFT (NASA) 161 p

N93-27058

Unclass

G3/71 0163199



MITRE

WORKING PAPER

Center for Civil Systems

WP 92W0000302

Title: Integrated Station Executive Requirements and System Design Approach

Author(s): Eugene L. Berger, C. Doug Morris

Dept.: H125

Project No.: 3100M

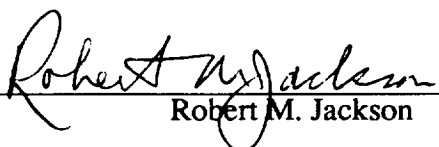
Date: November 1992

Contract No.: NAS9-18057

Issued at: Houston

Sponsor: NASA/JSC

Approved for MITRE Distribution:


Robert M. Jackson

ABSTRACT:

This document summarizes the current state of the Freedom Integrated System Executive (ISE) requirements and assesses the characteristics of the current design. MITRE's goals in this summary and assessment are two-fold: first, to identify any internal inconsistencies in either the requirements or in the current design; and second, to examine the applicability of the Open System Interconnection (OSI) management standards. Inasmuch as the ISE has been defined as the executive or operations manager application within the integrated avionics software of the space station, special attention is given to adapting OSI management standards for the specification of the ISE functions and the on-board Data Management System (DMS) services.

KEYWORDS: Freedom, Open System Interconnection, OSI management standards, Interface Control Documentation, international standards, management standards, Integrated System Executive, Data Management System

(NASA-CR-185708) INTEGRATED
STATION EXECUTIVE REQUIREMENTS AND
SYSTEMS DESIGN APPROACH (Mitre
Corp.) 235 p

N93-27143

Unclass

THIS INFORMAL PAPER

Department Approval:


Robert M. Jackson

MITRE Project Approval:

 8/10/92
Jack C. Heberlig 11/9/92

Peer Approval:


Claude E. LaBarre

EXECUTIVE SUMMARY

The Avionics Office of the Space Station Projects Office at Johnson Space Center (JSC) is working to define and integrate end-to-end requirements for the Space Station *Freedom* Program (SSFP) space-ground operations. As part of these efforts, the project office has had The MITRE Corporation perform assessments and analyses in areas where they had particular concern. These areas include the changing concepts for test methodologies, the operation and performance of the communication protocols, end-to-end network management, and the Master Objects Data Base (MODB). Since the recent restructure of the space station design, a new software application, the Integrated Station Executive (ISE), has been established. This application is to act as an executive agent along with the crew and ground controllers, while replacing (or absorbing) many of the system management functions that required a home when distributed element management was eliminated.

This document summarizes the current state of the ISE requirements and assesses the characteristics of the current design. MITRE's goals in this assessment and analysis is two-fold: first, identify any internal inconsistencies in either the requirements or in the current design; and second, to examine the applicability of the Open System Interconnection (OSI) management standards. Inasmuch as the ISE has been defined as the executive or operations manager application within the integrated avionics of the space station, special attention is given to adapting OSI management standards for the specification of the ISE functions.

In examination of the ISE requirements, the primary objective is the clarification of the purpose and design of the ISE. From the beginning of MITRE's involvement in this area, we have recognized that the development community has been hampered by the poor definition of the ISE requirements and performance characteristics. This problem has been made worse by the almost continual state of redesign, restructuring, and reestablishing priorities of the SSFP requirements. Definition of terminology within the program is constantly changing and the introduction of new terminology has led to further confusion. This problem is particularly acute in the area of object oriented programming and in the use of the terminology in the OSI management standards. This document will have served a very useful purpose to the community, if nothing else is accomplished, if it provides a reference dictionary of National Aeronautics and Space Administration (NASA) and OSI terminology in its description of the ISE design.

The technical approach taken in performing the assessment was to assemble the requirements for the ISE as described in the post-restructure versions of the SSFP design documents. Then the required ISE functionality was examined to develop a reference architecture to study the applicability of international standards for open systems to the ISE. This

architecture shows how the Data Management System (DMS) standard services (STSVs) are used by the ISE and how they fit into the open systems concept.

The OSI standards are found to be a close match to the needed specifications for the ISE functions and DMS standard services. The adoption of most clauses of the standards from the International Organization for Standardization/International Electrotechnical Commission ISO/IEC 10165 part 1, part 2, and part 4; ISO/IEC 10164 parts 1 through 6; and committee drafts ISO/IEC 10164 parts 7, 9, 11, 12, and 13 could provide the SSFP with a:

- Consistently defined set of operation management terminology
- Consistent operations management architecture
- Standard set of formatted messages
- Standard way of interfacing with the:
 - Space Station Control Center (SSCC)
 - Payload Operations and Integration Center (POIC)
 - International partners (IP)
- Standard way of documenting the:
 - Commands
 - Access controls
 - Behaviour
 - Attributes
 - Testing
 - Telemetry of on-board systems, elements, and payloads.

The contractual adoption of (or at least the establishment of contractor agreements to comply with) the international standards would help speed the delivery of the SSFP end-items by dramatically reducing the extent of interface documentation needed among *Freedom*, the SSCC, the POIC, and the IP.

The recommendations of the assessment of the ISE are as follows:

NASA should clarify and freeze the ISE requirements -- Modify the requirements to make them more explicit, rewrite the requirements to add the following basic functions:

- Add requirements for a Command Sequencer, Command Discriminator, and Scheduler
- Add requirements for a notification and event discriminators
- Add requirements to establish ISE's performance characteristics
- Add requirements to clarify ISE's resource management responsibilities

- Add requirements to clarify ISE's configuration management responsibilities
- Add requirements to clarify who has management responsibilities for journalizing, attribute scan list selection, and telemetry object list selection

NASA should accept and enforce the standard definitions of terminology and objects --
To help eliminate the confusion problems related to terminology and definitions:

- Establish and publish a single list of definitions across the entire SSF program
- Adopt as SSFP standards the ISO/IEC International Standards on OSI Management
- Establish a registration authority (configuration management) as soon as possible for SSFP managed objects and data objects

TABLE OF CONTENTS

SECTION	PAGE
1 INTRODUCTION	1
1.1 Background	1
1.2 Purpose of Document	2
1.3 Technical Approach	2
1.4 Structure of the Document	3
2 ISE FUNCTIONAL REQUIREMENTS	5
2.1 ISE Level A Functions	5
2.1.1 ISE Functional Requirements	6
2.2 ISE CEI Functionality	10
2.2.1 Station Mode Control	11
2.2.2 System Control	11
2.2.3 Secondary Power Control	14
2.2.4 Failure Reconfiguration	15
2.2.5 Caution and Warning Suppression	15
2.2.6 Caution and Warning Synthesis	15
2.2.7 Operational Plan and Execution Control	16
2.2.8 Payload Support	17
2.2.9 Rack Control	17
2.2.10 ISE Support to JEM and APM	17
3 SUGGESTED ISE ARCHITECTURE	19
3.1 ISE as a Tier 1 Peer	20
3.2 ISE as a Supporting Agent for the Crew and Ground Controllers	21
3.3 ISE Integrating and Coordinating Functions	24
3.3.1 Station Mode Manager	25
3.3.2 ISE Event Manager	27
3.3.3 ISE Systems Manager	29
3.3.4 ISE OSTP Manager	32
3.3.5 Command Sequencer and Scheduler	33
3.3.6 Event Discriminator	34

SECTION	PAGE
4 ISE SUPPORTING FUNCTIONS	37
4.1 Object Management Function	38
4.1.1 Object Management Model	38
4.1.2 Object Management Generic Notification Definitions	41
4.1.3 Object Management Service Definitions	45
4.1.4 Object Management Protocol and Abstract Syntax Definitions	45
4.2 State Management Function	47
4.2.1 State Management Model	59
4.2.2 State Change Notifications	60
4.2.3 State Management Service Definitions	61
4.2.4 State Management Protocol and Abstract Syntax Definitions	62
4.3 Attributes for Representing Relationships	62
4.3.1 The Model of Attributes for Representing Relationships	65
4.3.2 Notification of Changed Attributes that Represent Relationships	65
4.3.3 Attributes and Objects for Representing Relationships Service Definitions	65
4.3.4 Attributes for Representing Relationship's Protocol and Abstract Syntax Definitions	67
4.4 Alarm Reporting Function	68
4.4.1 The Model of the Alarm Function	73
4.4.2 Notifications of the Alarm Function	74
4.4.3 Attributes and Objects for Alarm Function Service Definitions	74
4.4.4 Attributes for the Alarm Function and Abstract Syntax Definitions	75
4.5 Event Reporting Function	76
4.5.1 The Model of the Event Reporting Function	81
4.5.2 Notifications of the Event Reporting Function	81
4.5.3 Attributes and Objects for Event Reporting Function Service Definitions	82
4.5.4 Event Function Protocol and Abstract Syntax Definitions	83

SECTION	PAGE
4.6 Objects and Attributes for Access Control Management	83
4.6.1 The Model of the Objects and Attributes for Access Control	85
4.6.2 Notifications of the Objects and Attributes for Access Control	94
4.6.3 Attributes and Objects for Objects and Attributes for Access Control Service Definitions	94
4.6.4 Protocol and Abstract Syntax Definitions of Objects and Attributes for Access Control	95
5 FINDINGS, RECOMMENDATIONS, TRADE-OFFS, AND RISKS	97
5.1 Findings and Recommendations Concerning the ISE Requirements	97
5.1.1 Findings	97
5.1.2 Recommendations	100
5.2 Trade-offs and Risks in Adopting ISO/IEC Standards	101
5.2.1 Object Management	101
5.2.2 State Management	101
5.2.3 Attributes Representing Relationship Management	102
5.2.4 Alarm Management	102
5.2.5 Event Management	103
5.2.6 Testing Management	103
5.2.7 Log Control Management	104
5.2.8 Objects and Attributes for Access Control Management	104
5.2.9 Summarization (TOL) Management	105
5.2.10 Scheduling Management	105
LIST OF REFERENCES	107
Appendix A STANDARD TERMINOLOGY AND DEFINITIONS	111
Appendix B OPEN SYSTEMS MANAGEMENT TUTORIAL	121
Appendix C MANAGEMENT SERVICE CONTROL STANDARDS	125
Appendix D EXAMPLE OF GDMO	131
Appendix E PROPOSED COMMAND SEQUENCER AND DISCRIMINATOR OBJECT CLASS DEFINITIONS	137

SECTION	PAGE
Appendix F PROPOSED SPACE STATION OBJECT AND RELATIONSHIP ATTRIBUTES	173
Appendix G LOG CONTROL FUNCTION	185
Appendix H SUMMARIZATION FUNCTION (Telemetry Object List Management)	195
Appendix J THE TEST MANAGEMENT FUNCTION	207
Appendix K THE SCHEDULING FUNCTION	219
GLOSSARY	227
DISTRIBUTION LIST	231

LIST OF FIGURES

FIGURE	PAGE
1 The Tier 1 Model for Commanding of the Freedom Systems, Elements, and Payloads	21
2 The ISE as a Supporting Agent to the Crew and Ground Controllers.	23
3 ISE Functional Architecture Diagram	26
4 Operational State Model for Managed Objects	48
5 Usage State Model for Managed Objects	50
6 The Administrative State Model for Managed Objects	52
7 The Event Report Management Model	77
8 The Generic Access Control Model	88
9 Access Control During Association Establishment	89
10 Access Control During Management Operations	90
11 Management Notifications with Applied Access Controls	91
12 Basic Management Framework	123
13 The Management Service Control Function	127
14 ISO Access Control via DMS STSVs - Action I/O	129
15 The Simple Programming Constructs	138
16 Sample Command Sequence	142
17 The Discriminator Model	146
18 The Command Sequencer Model	158

FIGURE	PAGE
19 Summarization Objects Observing Attributes Within Objects Under Observation	199
20 The Test Function Model	210
21 The Synchronous Test	211
22 The Asynchronous Test	212
23 The Scheduling Function Model	221

LIST OF TABLES

TABLE	PAGE
1 ISE System Control Functions versus Freedom Systems and Elements	31
2 Parameters for the State Change Notification	60
3 Parameter of the Changed Attributes of a Relationship Change Notification	66
4 Sample Sequence	139
5 Example Command Sequencer -- Initialize_CMG	140
6 Example Command Sequencer -- Power_and_Test_CMG	140
7 Example Discriminator -- ON_ERROR	141
8 Example Discriminator -- Verify_Heater 1 ON	141
9 Example Discriminator -- Heater_Failure	141

SECTION 1

INTRODUCTION

1.1 Background

The National Aeronautics and Space Administration (NASA) Avionics Office of the Space Station Projects Office at Johnson Space Center (JSC) is working to define and integrate the end-to-end requirements for the Space Station *Freedom* Program (SSFP) space-ground operations. As part of these efforts, The MITRE Corporation performed assessments and analyses for the program office in the following areas: the evaluation of the changing concepts for test methodologies, the operation and performance of the adopted communication protocols, the evaluation of the end-to-end network management, and the operations concept for the Data Management System (DMS) Master Object Data Base¹ (MODB).

Since the recent restructure of the space station design, a new software application, the Integrated Station Executive (ISE), has been established. This application is to act as an executive agent along with the crew and ground controllers, while replacing (or absorbing) many of the system management functions that required a home when distributed system management was eliminated.

As a natural follow on to MITRE's current activities, the Avionics Office has asked MITRE to provide a technical assessment of the requirements and the design of the ISE and its relationship with the ground control, the crew, and the DMS. As part of this effort, MITRE will identify and characterize issues related to the ISE implementation and design and provide technical assessment and recommendations as necessary. This document is a summary of MITRE's efforts in this area.

¹ DMS objects: A DMS data object is an abstract representation of the storage spaces for attribute values. DMS objects can either be read only or read-write. DMS data objects do not have any operational management behaviour. The behaviour of on-board managed objects are part of the hardware characteristics, the environment, and the associated software application. The behaviour of the managed objects is the result of changes to the object attribute values, changes to the the environment, and changes to the behaviour rules of the physical and logical managed objects.

1.2 Purpose of Document

This document provides an analysis of the current state of the ISE requirements and includes an assessment of the characteristics of the current design. The emphasis in this analysis is two-fold: first, to identify any inconsistencies in either the requirements or in the current design; and second, to examine the requirements and the design for the applicability of the Open System Interconnection (OSI) management standards. Since the ISE has been defined as the executive or system manager application, special attention is given to the OSI standards for management functions and the applicability of these standards for fulfilling the needs of the ISE.

The primary objective of MITRE's examination of the ISE documentation is to clarify the purpose, requirements, and design of the ISE. From the beginning of MITRE's involvement in this area, we have recognized that the development community has been hampered by the poor definition of the ISE requirements and performance characteristics. This problem has been aggravated by the almost continual state of redesign, restructuring, and re-establishing priorities of the SSFP requirements. Definition of terminology within the program is constantly changing, and the introduction of new terminology leads to further confusion. This problem is particularly acute in the area of the object model and in the use (or misuse) of the terminology established by the OSI management standards.

This document provides a current system level "snapshot" of the ISE design and how it fits into the overall station integrated avionics and the command and control structure. Design decisions for the ISE cannot be made independently from those made for the DMS and its support functions. Further, the design decisions for the ISE cannot be made independently from those made for the other on-board system, element, and payload applications. The details of the relationships between ISE and the applications it manages and the services it employs must be understood and documented. This document intends to set up a reference architecture for the ISE to clarify the current ISE design and guide future ISE design decisions. This reference architecture will attempt to show how well the ISE maps to (or agrees with) the OSI management standards and how the DMS STSVs help ISE accomplish its management functions.

1.3 Technical Approach

A standard systems analysis approach was incorporated in the preparation of this study and report. The system level documentation was examined and the ISE requirements were identified and validated. Validation of the requirements was accomplished through intensive review of needs at all levels of both NASA and the contractor communities. This validation process is an on-going effort and will continue through the establishment of the level "C" requirements currently. The currently identified ISE requirements have been further

examined in the context of the DMS support capabilities and in the context of ISE's role within the integrated avionics environment. Applicable OSI standards were then researched to identify their applicability to the ISE requirements. A reference architecture for the ISE was then established and necessary supporting functions were defined. The proposed ISE architecture, compliant with OSI standards for systems management, can now be used as a tool to examine the evolving functional design of ISE.

1.4 Structure of the Document

In section 2 the requirements and functions of the ISE are presented as they exist in the post-restructure versions of the SSFP design documentation.

Section 3 examines the required ISE functions and performance characteristics and discusses an architecture for ISE that is compatible with open systems standards for systems management.

Section 4 provides a detailed presentation of a suite of support functions for the suggested architecture discussed in section 3.

Section 5 summarizes this document by providing a discussion of the findings, recommendations, design trades, and risks associated with the ISE, the DMS standard services, and applications of ISO standards to the management functions.

In all of the above discussions, the document will make use of a consistent set of terminology based on the usage in the International Organization for Standardization (ISO) literature. Careful attention has been given to make the reader aware of any identified conflicts between current SSFP terminology usage and the terminology in the standards. New terminology is introduced from the standards only where necessary or where the SSFP uses are inconsistent or incomplete.

Appendix A provides a list of the standard definitions and terms used in this report. It is strongly recommended that NASA consider this list as a starting point for standardizing terminology throughout the SSFP.

Appendix B presents a brief tutorial that covers the ISO OSI management standards, the scope and content of the standard management functions, and their relationship to the ISE design.

Appendix C presents a brief discussion of ISO/IEC standard models for management service control (MSC) functions and how they might be applied to meet the ISE requirements.

Appendix D provides an example of the use of the ISO/IEC Guidelines to Define Managed Objects (GDMO) to collect the information necessary to define and register management objects². This example is applicable to all SSFP managed objects.

Appendix E includes a proposed design for two management support object classes seen as critical to the design of the ISE. The object classes defined are the Command Sequencer and the Command Discriminator.

Appendix F proposes a standard set of object attributes and object relationship attributes to be used by all other on-board ISE managed applications. The use of a standard set of such attributes will eliminate duplication of effort and minimize confusion between application developers in designing the interfaces with ISE. This appendix is an expansion of the standard definitions needed by the SSFP development community.

Appendix G includes a description of the ISO Standard on the log control function and how it may be applicable to *Freedom*.

Appendix H includes a description of the ISO committee draft standard on the summarization function and how it may be applicable to *Freedom* telemetry object list management.

² Managed objects: A managed object is an abstract representation of resources of a managed system. The management of these resources requires a management view of the logical and physical identities within the managed system. Managed objects have behaviour and the managed object behaviour is the result of either changing attribute values, commanding actions to change the managed process, changing the managed object's environment, or the behaviour rules associated with the managed object. Examples of managed objects are systems, elements, payloads, orbital replaceable units, standard data processors, mass storage units, etc.

SECTION 2

ISE FUNCTIONAL REQUIREMENTS

This section presents a description of the functions partitioned to the ISE as a result of the activities since the SSFP software restructure scrub. Two reference documents have been used to compile this picture of the ISE:

- *Level A Integrated Flight Software Architecture Requirements Document Section 1 Parts 1 and 2*, (NASA, August 15, 1991, [SSP 30555]),
- *Contract End Item (CEI) Specification for the Data Management System, vol. 2 Integrated Station Executive*, (MDSSC, September, 1991[DR SY-06.1]).

MITRE notes with concern that these two documents represent completely different views of the ISE requirements. *Level A* presents high-level, generic ISE capabilities while the *CEI* describes distinct flight software functions. As such, the ISE requirements in the two documents are often inconsistent and traceability from *Level A* to the *CEI* is often difficult and sometimes impossible. To provide a complete picture of the ISE requirements, the discussion in this section includes descriptions of the ISE functions from both documents' perspectives. A third document, *Software Restructure Scrub - Summary of Results*, (Dawson, Whitelaw, 1991) has also been used to help clarify some of the many issues that remain vague in the two ISE documents.

2.1 ISE Level A Functions

The ISE is the station executive which provides the capability for the crew and ground to perform centralized and real-time integrated command, control, and management of the Space Station *Freedom* operations. The ISE will be accessible from any workstation with the User Support Environment (USE) capability. The ISE functionality is derived from a combination of application software and operational data and is supplied through the use of standard DMS data and command services. These data and command services include the Intermediate Language Executor (ILE)³ and access to the Run-Time Object Database

³ Intermediate Language Executor (ILE): specific use of this DMS service and its existence in the program are the subject of debate at this time. Level A requirements here must be considered subject to change. For a detailed description of the ILE and UIL see SSFP UIL Specification, USE 1001, 15 March 1990.

(RODB). The ISE uses the DMS supplied data networks to provide command and control connectivity among the on-board applications.

These capabilities allow the ISE to perform real-time and near-real-time management of station operations requiring coordination among systems and elements. The ISE will be capable of monitoring and controlling the station's systems, elements, and payloads during both manned and unmanned operations and will perform control functions in response to the flight crew and ground controller commands.

The ISE will issue commands to configure the station equipment and resources by executing station and system level operational procedures. The ISE provides the capability to execute predefined stored command sequences as responses to crew or ground commands or as the result of predefined conditions or events.

The ISE provides a centralized interface for the crew and ground for mode, configuration, activity, event, and timeline control. The ISE uses the services of both DMS and Communications & Tracking (C&T) to establish and maintain a communications link with the SSCC. In support of ground control operations, the ISE will provide audit trails of the interactions between ISE and applications within the station's systems, elements and payloads.

2.1.1 ISE Functional Requirements

The ISE consists of code and supporting data such as executable timelines, procedures, displays, and other data deemed necessary for reconfiguration operations. The ISE uses this data to accomplish the requirements of coordinated commanding, the monitoring of operational status, the control and execution of the on-board short term plan (OSTP), and the synthesis of Caution and Warning events. In concert, the ISE will provide the following integrated functions:

- Mode Management
- Configuration Management
- Activity Management
- Event Management
- Timeline

Each of these integrated functions are discussed in the following paragraphs.

2.1.1.1 Mode Management

The ISE mode management function provides the capability to reconfigure *Freedom* in support of its various operational modes. The ISE executes integrated station mode transition procedures in response to crew or mission controller commands, as scheduled in a timeline or in response to an event. The integrated mode transition procedures contain sequences of commands required to establish the new mode's constraint environment and may include: precondition checks, commands that set or release interlocks⁴, command inhibits⁵, and command enables⁶, and system configuration information. The ISE executes the procedures to effect the transition from the current station mode to the commanded station mode. As part of this station mode transition process, the ISE may command systems, elements, and payloads to compatible modes and configurations. The ISE will then enforce the station mode environment by establishing access controls that allow only those commands that are compatible with the current station mode. The ISE will monitor system, element, and payload status for compliance with the current station mode and will notify the crew and ground of mode violations. The crew and the ground will have the capability to override any ISE established moding constraints. As necessary, the ISE will have the capability to add or delete station modes.

NASA has defined station modes to provide flexible management of *Freedom* operations so that safe interactions between the distributed systems, elements, and payloads can be maintained. Station modes are characterized by permissions and restrictions imposed upon all of the station operations. Each mode establishes a framework within which operations, functions, and activities can be performed during a given time frame. If station operators desire an operation, function, or activity that is incompatible with the current station mode, a

-
- ⁴ Interlock: a managed object that is an extra hardware or software control used to ensure additional operational safety. Interlocks provide functionally independent control over a process or a device that may be hazardous or disruptive. Interlocks have behaviour that require a two-step command sequence.
 - ⁵ Command Inhibit: the temporary removal of the ability to receive commands from a device or software application . An external inhibit blocks the command path to a device. An inhibit may also be the prohibition of command execution placed on a device or software application. An internal inhibit prevents the software from executing commands.
 - ⁶ Command Enable: providing the authority to the execute commands by a device or an application.

transition to the appropriate mode or an override by the flight crew or mission controllers is required.

Seven station modes have been defined. This set of station modes may be redefined, expanded, or reduced in response to station capabilities or operational objectives. The current definitions for each of the modes are presented below:

Maneuver mode: the maneuver mode permits operations supporting propulsive maneuvers except for Proximity operations.

Micro-gravity mode: this mode supports micro-gravity experiments and quiescent payload operations.

Normal mode: this mode permits operations to support most of the station operations.

Proximity operations mode: this mode allows operations to support the control and interaction with vehicles within the station Command Control Zone. This mode also supports docking and berthing.

Safe mode: this mode is dedicated to providing flight crew safety and vehicle integrity. The intent of this mode is to support recovery from station-level emergency conditions. All unnecessary operations are terminated and the remaining station resources are dedicated to providing long-term flight crew life support.

Transfer mode: the transfer mode is oriented toward the transfer of flight crew, materials, resources, and payloads. This mode shall support EVA, special maintenance, and assembly operations. It also shall support all vehicle attached operations.

Unmanned mode: this mode is intended to support primarily assembly sequences and assumes that a crew is not present on the station. The unmanned mode supports operations of the station as commanded from the ground or remotely from the orbiter.

Aside: It is noted that these mode definitions are often overlapping in scope and inconsistent in content. For example, as they are currently defined by NASA, the station cannot be both unmanned and in the safe mode. Another example - the station cannot be in the unmanned mode and in the microgravity mode. It is recommended that the station modes be redefined not to be exclusive. NASA is currently attempting to refine its position on the mode concept and its applicability to station design and operations.

2.1.1.2 Configuration Management

The ISE configuration management function provides the capability to reconfigure *Freedom* as needed, upon crew or ground command, in response to predefined conditions, or as

contained in the OSTP. The ISE shall also provide for reconfiguration commands for resource distribution.

2.1.1.3 Activity Management

The ISE activity management function provides for the coordinated commanding of *Freedom* components to accomplish mission objectives. When executing an activity, the ISE may issue commands, including pre-stored program sequences, to systems, elements, and payloads. The ISE will verify conditions before and after executing an activity and will provide the capability to display the status of activity execution. The ISE will monitor specified system, element, and payload status to ensure compatibility with executing activities. The ISE will allow the crew and the ground the ability to control the activities (e.g., execute, terminate, suspend, and resume) and will provide the capability for the crew to select between manual, single step, and automatic control of activity execution.

2.1.1.4 Event Management

The ISE event management function provides for an automated response to predefined events. The ISE will detect and respond to predefined events involving more than one system, element, or payload. The ISE will detect events by:

- Receiving event notifications from systems
- Receiving Caution & Warning messages from systems
- Receiving manually-input event declarations
- Monitoring station configuration and operational status

The ISE will respond to an event by:

- Issuing a predefined integrated sequence of commands to the system that isolates the failure
- Issuing a command sequence designed to support failure recovery
- Generating C&W messages when specified patterns of system events and/or C&W messages have been detected
- reporting the event to the crew and ground controllers

2.1.1.5 Timeline Management

The ISE Timeline management function provides management of a coordinated schedule of activities. The ISE will execute scheduled activities in an integrated environment. The ISE will provide the capability to display resource availability, consumption, and environmental

parameter disturbances for all activities scheduled on the timeline. The ISE will assess the timeline for resource rights and privileges conflicts at the request of the crew or mission controllers. The ISE will have the ability to compute and display projected start and finish times for scheduled activities. Finally, the ISE will provide the capability to the crew and ground to add, delete, and modify the timeline of a single scheduled activity in real time.

2.2 ISE CEI Functionality

Since the station software scrub activity, the WP-2 contractor has been directed to modify the software documentation tree and to document flight software according to the Flight Software System Requirements (FSSR) data item description (DID). In essence, this new document combines the previous version of the ISE Contract End Item Specification (CEI), the various ISE IRDs and ICDs, the ISE SRS, and a number of other contracted documents into one document. In addition to the combined scope of the ISE FSSR, the document presents a significantly different view of the ISE requirements than that given in *Level A*. The current ISE FSSR, however, only covers requirements through the MTC phase of the program. The ISE *CEI* specification has been updated to reflect the FSSR functions, while covering the requirements through PMC. Therefore, to maintain continuity with the ISE *Level A*, the ISE *CEI* specification was selected to describe the WP-2 vision of the ISE requirements.

The ISE, a single computer software configuration item (CSCI), is a part of the integrated avionics system of the *Freedom*. The ISE performs as the executive software application and is used by the crew and authorized ground personnel to command, control, and manage the Space Station *Freedom*. The ISE is the only on-board application that may send commands to other on-board applications. Consequently, the ISE plays a key role in the control of the station systems, elements, and payloads.

The ISE consists of the following principal functions:

- Station Mode Control
- System Control
- Secondary Power Control
- Failure Reconfiguration
- Caution and Warning Suppression
- Caution and Warning Synthesis
- Operations Plan Execution Control
- Rack Control
- Payload Support

- Japanese Experiment Module (JEM) and Columbus Attached Pressurized Module (APM) support.

The following paragraphs provide a brief overview of these functions as they are described in the ISE CEI specification.

2.2.1 Station Mode Control

The ISE performs transitions between station modes. The ISE establishes the environment defined by the new station mode by commanding inhibits, enables and interlocks that preclude incompatible operations from being executed. These mechanisms enforce the environment defined by the station mode. The crew and ground controllers can override these mechanisms.

In performing a station mode transition, ISE performs a set of tasks that control the mode transition. The set of tasks performed by ISE include:

- Receiving the command to transition to the new station mode
- Verifying a predefined set of initial condition checks
- Establishing a station-wide set of interlocks and inhibits
- Commanding systems to modes or configurations for target modes
- Performing a predefined set of post-condition checks
- Notifying crew and ground of completion of the transition

The ISE enforces the station mode by inhibiting those commands that are incompatible with the current station mode. In controlling systems, the ISE shall only transition systems to modes compatible with the active mode, unless overridden by the crew or the ground. The ISE will notify the crew when an ISE command is incompatible with the current station mode.

2.2.2 System Control

The ISE is responsible for the configuration control of the SSFP system. The ISE principal function of System Control interfaces with the following on-board systems and elements :

- Data Management System (DMS)
- Communications & Tracking (C&T) System
- Guidance Navigation & Control/Propulsion (GN&C/P) System
- Extravehicular Activity System (EVAS)/ Airlock

- Electrical Power System (EPS)
- Mobile Transporter Element (MTE)
- Mobile Servicing Center (MSC)
- External Thermal Control System (ETCS)
- Internal Thermal Control System (ITCS)
- Environment Control Life Support System (ECLSS)
- Man Systems/Crew Health Care System (CHeCS)
- Rotary Joint (RJ)
- Station Docking Mast Controller (SDMC)
- Unpressurized Logistics Carrier/ Propulsion Module Attachment (ULC/PAMS)

The System Control function of ISE issues coordinated commands to systems through the auspices of five constituent capabilities. These capabilities are system initialization or activation, system shutdown or deactivation, system moding, system operational reconfiguration, system test and checkout. The scope and role of these ISE constituent capabilities will vary for each system and will be defined by the system and subsystem designers. The general scope of the capabilities are discussed below.

To perform their defined role within the ISE, each constituent capability issues commands that:

- Change system modes
- Switch system TOLs
- Power on/off Orbit Replaceable Units (ORUs)
- Notify systems of ORU availability
- Inhibit and enable the execution of system commands
- Apply and remove station mode related interlocks
- Switch Multiplexer/Demultiplexer (MDM) Input/Output Data Base (IODB) scan lists for systems located in MDMs
- Affect system operations as pre-approved

2.2.2.1 System Initialization or Activation

The ISE constituent capability role in system initialization works closely with DMS initialization. The ISE will verify a predefined set of initial conditions and then send commands to connect power to DMS hardware components in an avionics string. DMS Station Manager (SM) will then load the system application software for each component power on. For system activation the ISE constituent capability will send commands to connect power to system ORUs on an avionics string. The ISE will then verify a predefined

set of initial conditions and then send commands to establish a system's operational state as active.

The system activation capability connects power to system ORUs on an avionics string and commands the system's software to establish its operational state. The system software will then be commanded to establish the necessary system and/or subsystem modes.

2.2.2.2 System Shutdown or Deactivation

The ISE constituent capability role in system shutdown is to disconnect power from DMS hardware components on an avionics string. This capability includes powering off the SDP and MSU, and the nominal set of MDMs and notifying the DMS of ORU non-availability.

The system deactivation constituent capability includes sending commands to disconnect power from a system's ORUs on an avionics string. The ISE will send commands to the system application software to establish a system non-operational state. The ISE will also support commanding system and subsystems to the appropriate system mode.

2.2.2.3 System Moding

System mode control commands are executed by the ISE in conjunction with its role of maintaining the individual systems in the proper mode state for the current station mode. The ISE will execute system mode transitions as they are commanded by either the crew, authorized ground controllers, as they are scheduled in the on-board short-term plan (OSTP), or as they may be required by a predefined response to a Caution and Warning notification. The steps performed by the system moding subfunction in transitioning systems from one mode to another are:

- Verify commanded mode is possible before commanding transition
- Verify target system mode is compatible with current station mode
- Verify initial conditions
- Connect and disconnect power to systems ORUs
- Notify systems of ORU availability
- Command DMS to switch TOLs
- Activate command enables, inhibits, and interlocks
- Command system to target mode

2.2.2.4 System Operational Reconfiguration

For system operational reconfiguration, the constituent capability will, after verifying a set of initial conditions, configure systems to provide for redundant strings for time- and safety-critical station operations. System ORUs can be powered on or off, on a string by string basis. This capability shall also notify systems of ORU availability and shall command DMS to switch TOLs as necessary.

2.2.2.5 System Test and Checkout

System Test and Checkout Control command sequences are issued by the ISE to support the accomplishment of system test and checkout. The constituent capability will verify a predefined set of initial conditions, configure systems for a test, and issue predefined test commands.

The system test and checkout subfunction configures systems for the crew and ground to perform a system test. System ORUs can be powered on or off and systems will be moded to an appropriate test mode. The ISE will notify systems of ORU availability and activate command inhibits, enables, and interlocks as required.

2.2.3 Secondary Power Control⁷

The ISE provides centralized control over secondary power distribution for the space station. The ISE controls secondary power distribution by issuing commands to open and close switches within Remote Power Controllers (RPCs). The ISE receives commands to control power to station ORUs.

The invocation of ORU power commands can be in response to a command from the crew, from authorized ground controllers, and from scheduled procedures in the OSTP. Upon the receipt of a power on or off ORU command, the ISE issues a command to open or close the appropriate RPC.

The ISE monitors data from the RPCM describing the current RPC configuration and reports to the crew and ground any detected state changes.

⁷ The ISE function with regard to secondary power is currently under review by NASA and is expected to change. Many of the requirements included in earlier ISE documentation has been reduced or eliminated in the contractor's latest versions of the ISE FSSR.

2.2.4 Failure Reconfiguration⁸

The ISE provides a failure reconfiguration capability to respond to predefined failure notifications that require cross-system commanding capabilities unique to a Tier I entity. The failure reconfiguration processing performed by ISE begins with the receipt of one or more event notifications from system or element software via DMS services. The ISE then identifies the failure that has occurred (with a predefined failure type) and the reconfiguration action to respond to the failure. The ISE will then initiate the identified reconfiguration action. The actual reconfiguration action is carried out by the ISE principal function of System Control. Individual reconfiguration actions can be inhibited or enabled by commands from the crew or ground.

Changes to the data used by ISE to identify and respond to event notifications received from other systems or elements can be changed by loading new reconfiguration data from the ground. The ISE will notify the crew and the SSCC of all automatic reconfigurations undertaken and of all reconfigurations that could not be completed.

2.2.5 Caution and Warning Suppression

When predefined Emergency, Warning, and Caution alarms are reported to the ISE, the ISE has the ability to suppress the annunciation of subsequent Caution and Warning messages. In this way, the ISE prevents nuisance alarms and message floods. The ISE performs the C&W suppression function by associating a predefined set of Caution and Warning alarms with each Emergency, Warning or Caution alarm received. When an alarm is received that has a predefined set of associated alarms, the ISE commands the DMS to suppress annunciation of the alarms in the associated set. This ISE alarm suppression function is only active with prior crew or ground authorization. This ISE function will accept new C&W suppression criteria from the ground.

2.2.6 Caution and Warning Synthesis

The caution warning synthesis principal function of ISE synthesizes new C&W messages by recognizing patterns of C&W messages generated by other sources. The augmentation of the DMS capability is derived from ISE's station-wide perspective of systems, elements, and payloads. The ISE uses the DMS STSV C&W processing service to annunciate its

⁸ As with the discussion on secondary power control in the previous paragraph, ISE failure reconfiguration function has been significantly reduced in scope from that contained in both previous WP-2 ISE documents and from that contained in the *Level A* for ISE. See footnote 7.

synthesized C&W event alarms. The ISE provides the capability for the crew and ground to modify the criteria on which it bases its C&W synthesis.

2.2.7 Operational Plan and Execution Control

The ISE as the executive application of the integrated on-board avionics is the application responsible for the execution of the OSTP. The OSTP contains the information necessary to allow the ISE to execute automated procedures associated with scheduled activities. The ISE gives the crew and ground controllers the capability to modify the OSTP *in real time*. The ISE also allows the crew to assess any unexecuted portion of the OSTP. During an assessment, the ISE identifies resource consumption conflicts and environmental privilege conflicts. The ISE makes all of this data available to the crew for display and for downlink to the SSCC. The ISE also includes the capability to compute projected start and finish time for activities on the OSTP timeline.

The ISE uses the OSTP timeline data to control the execution of activities at scheduled times. The ISE can terminate and resume execution of the OSTP timeline on command from either the crew or authorized ground controllers. The ISE also provides the capability to incorporate new versions or revisions to the OSTP based on commands from the crew or SSCC controllers. Some activities on the station Master Activity Plan contained in the OSTP requires crew interaction. The ISE will prompt the crew for input for those selected activities. Normally the ISE will directly execute activities without requiring modification from either the crew or the SSCC. In contingency operations, the ISE can be commanded to incorporate predefined alternative activities into the OSTP on command from the crew or the ground.

In the execution of procedures for scheduled activities, the ISE verifies that all preconditions specified by the OSTP are satisfied prior to beginning the execution of an activity and that all post-conditions specified by the OSTP are satisfied prior to the termination of the activity. Pre- and post-conditions examined by the ISE include the proper interlock statuses, the required status of all station, facility, and crew, and the proper termination status of all prior activities necessary for the current activity to begin.

In the execution of an activity, the ISE allows the crew or SSCC to select between manual, single step, and automatic execution modes of activities. The ISE allows the crew or the SSCC to terminate, pause, and resume the execution of an activity. The ISE also provides

the capability to abort any currently executing activity on command from the crew or SSCC.⁹

The ISE provides the capability to the crew and the SSCC to edit the OSTP. The edit functions included in this capability allow for the addition or deletion of an activity from the timeline. The edit function also permits scheduled OSTP activities to be modified and it permits the crew or ground personnel to change the start time and duration of an activity scheduled for execution in the OSTP.

2.2.8 Payload Support

The ISE issues coordinated commands to support payload control by executing predefined command sequences. Upon command from an authorized source, the ISE issues two categories of payload support command sequences:

- Payload secondary power control
- Hazardous Payload Command and Control

The ISE will provide and remove power to a payload ORU upon command or upon detection of a predefined event. The ISE also has the ability to execute a command sequence to potentially hazardous payloads on command or upon detection of a predefined event.

2.2.9 Rack Control

The ISE provides for rack initialization and shutdown. The ISE coordinates secondary power, thermal cooling, and avionics air as part of its rack control responsibilities. Upon command from any authorized command source, the ISE will perform rack initialization or shutdown by sending commands to control secondary power distribution, ITCS, and to ECLSS.

2.2.10 ISE Support to JEM and APM

The ISE serves as the integrating function between station core systems, JEM subsystems, and APM subsystems. The exact scope and role of ISE in this interface remains a subject of international negotiations to be summarized in SSP 42001 (for APM) and SSP 42002 (for JEM.) ISE will be capable of sending commands and receiving data as specified in section 3.6 of the above referenced documents. In addition, it is also understood that ISE (or the ISE

⁹ The ISE *CEI* does not define these terms. It is, therefore, not clear how a terminate activity differs from an abort activity command.

host) will be responsible for providing station health and status data to JEM and APM. ISE's role in this gateway function is not yet defined. The commands listed below are those under consideration for ISE in the current version of the ISE *CEI* specification.

When properly directed from an authorized source, the ISE will issue the following commands to the JEM - Element Manager (EM):

- Select the mode of the JEM and JEM subsystems
- Initialize and shutdown the JEM and JEM subsystems
- Power on or off JEM ORUs
- Notify JEM-EM of ORU availability
- Command JEM-EM to enable or inhibit JEM system and subsystem functions
- Command the operations of JEM subsystem functions
- Notify the JEM-EM of the telemetry data to be downlinked via S-band

When properly directed from an authorized source, the ISE will issue the following commands to the APM - System and Mission Management (SMM) software:

- Select the mode of the APM and APM subsystems
- Initialize and shutdown the APM and APM subsystems
- Power on or off APM ORUs
- Notify the SMM software of ORU availability
- Command SMM to enable or inhibit APM subsystem functions
- Operate APM and APM subsystem functions
- Notify the SMM software of the telemetry data to be downlinked via S-band

SECTION 3

SUGGESTED ISE ARCHITECTURE

The current set of requirements for the ISE continue to be in a state of flux. Many aspects of the relationship between the ISE and the *Freedom* integrated avionics have not been defined or agreed upon. This is particularly true in the area of the Command and Control architecture of the *Freedom* and the ISE.

The following paragraphs describe an architecture for the ISE and its relationship with the DMS support functions and other on-board application software. The architecture described is an extension of that contained in the current ISE design documentation. The extensions described are presented as an example of a possible ISE implementation that encompasses the guidelines of the ISO open system standards for the management of SSF application software.

The purpose in developing this architecture description is to provide the ISE development community with a tutorial example of the scope of the ISO standards and the applicability of the standards to the practical design of the ISE software. It is not the intent of the authors to force a design for the *Freedom* ISE. We offer it as an aid to help design decision makers understand that incorporation of the ISO standards could make the final design easier to achieve while minimizing the implementation risks and life cycle costs.

The Space Station *Freedom* is currently defined to have a hierarchical, distributed command and control architecture. This architecture has as its highest level, called Tier 1, the functions that are concerned with global, station-wide issues. All the station-wide operations management functions take place at the Tier 1 level. The elements of Tier 1 are the crew, the ground controllers, and the ISE. The ISE is an on-board software application that operates in two roles: as a peer to the crew and ground controllers and as a supporting agent to the crew and ground controllers. In this document these two roles are called the Tier 1 ISE peer and the ISE supporting agent, respectively. The ISE is considered a Tier 1 peer to the crew and ground controllers when it executes commands either from the on-board short-term plan (OSTP) or from stored program command procedures (SPCPs) and performs station-wide integrating and coordinating functions. The ISE is considered a supporting agent when it executes commands for the crew or ground and performs its integrating and coordinating functions for the crew or the ground controllers.

3.1 ISE as a Tier 1 Peer

The ISE is the on-board Tier 1 software managing application that integrates and coordinates the execution of stored commands received via communications with the Data Management System (DMS) standard services from the OSTP or other sources of command sequences. As a managing peer, the ISE integrates *Freedom* by managing the configuration of the states of *Freedom's* modes, by checking *Freedom's* resource constraints, and by commanding operations invoked by discriminating *Freedom's* system C&W and FDIR messages. As a managing peer, the ISE coordinates the on-board systems, crew and ground controllers by sending commands to the on-board systems and messages to the crew and ground controllers.

As a managing peer, the ISE has its own objects that control its processes. These objects are the highest level objects on-board *Freedom*. These objects contain the attributes used to manage the scheduling of ISE behaviour, the establishment of priorities, the modes of *Freedom*, the selection of OSTP elements and delta changes to the OSTP, the selection of telemetry information, and the discrimination of *Freedom* system C&W and FDIR messages, and the commanding of procedures both to change the station operational modes and to implement failure isolation and recovery procedures.

In normal operation of *Freedom*, the ISE Tier 1 peer executes its station-wide integrating and coordinating functions. If the ISE Tier 1 peer cannot operate because of a failure, then the crew and ground controllers can directly command and control *Freedom's* systems, elements, and payloads.

Figure 1 illustrates the operation of the ISE as a Tier 1 peer. The ISE manages the on-board systems, elements, and payloads by sending commands to the managed system, element, or payload. The system, elements, and payloads respond to the commands and send response messages to the ISE. The system, elements, and payloads control the managed objects and use DMS services by reading and writing current values of control attributes to the run-time object database (RODB). The services performed for the systems, elements, and payloads are Management Service Control (MSC), C&W alarm message generation, management summary report generation of telemetry object list (TOL), access control, and the management of the configuration.

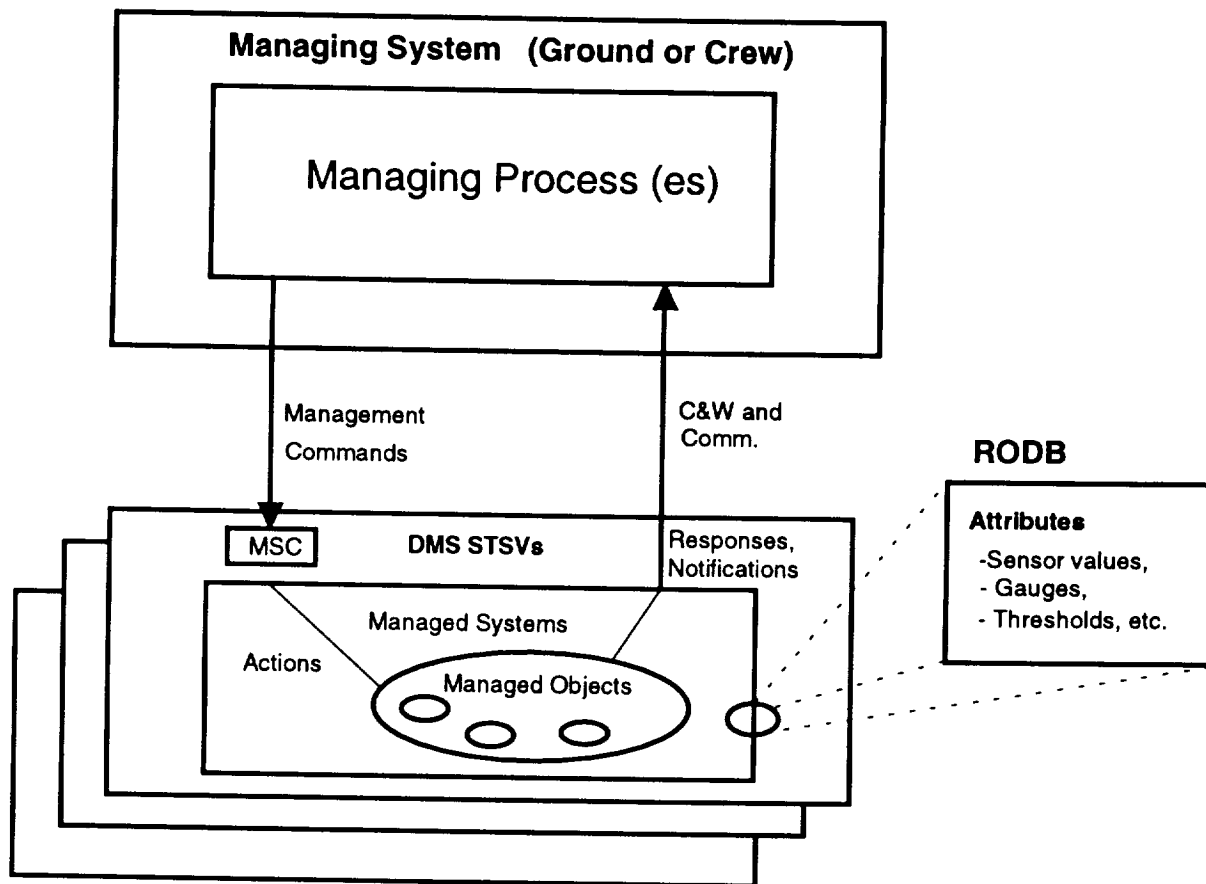


Figure 1. The Tier 1 Model for Commanding of the *Freedom* Systems, Elements, and Payloads

3.2 ISE as a Supporting Agent for the Crew and Ground Controllers

In normal operations, ISE is a software application that is managed by the crew and ground controllers and supports the crew in its management of the operation of the space station. In this role, the ISE operates as a supporting agent when the crew or the ground controllers send commands to *Freedom*. As a supporting agent, it responds to commands from the crew and the ground controllers, and it performs the station-wide integrating and coordinating functions. As a supporting agent, the ISE uses the DMS STSVs to perform the following functions:

- Reading and updating the current values of ISE's object attribute values from the run-time object database (RODB)
- Reading operational commands from the OSTP
- Reading and writing pattern recognition data for events and notifications
- Providing a real-time clock reference
- Providing communication service via the Network Operating System (NOS) for sending and receiving messages
- Checking the inhibit and enable access to the station's systems, elements, and payloads
- Providing caution and warning alarms and message generation and reporting to the crew and ground controllers
- Reporting of summaries of ISE objects' states and attributes (telemetry reporting)
- Logging of on-board commands
- Journalizing (checkpointing) of ISE system data

The crew and the ground controllers can by-pass the ISE supporting agent and send commands to *Freedom's* systems, elements, and payloads. In this case, the crew or the ground controllers perform the station-wide operations management functions and send commands directly to *Freedom's* systems, elements, and payloads. The DMS STSVs detects the received messages, updates the appropriate system attributes, and then DMS STSVs sends the message to the appropriate system, element, or payload.

Figure 2 illustrates the ISE as a supporting agent. The crew and ground controllers with their supporting software and hardware command the ISE to perform its managed processes. The ISE responds to the commands from the crew or ground controllers by sending responses to the commands. The ISE controls information objects and performs its managed processes. The ISE uses the DMS STSVs function as any other on-board system operating from a Standard Data Processor (SDP).

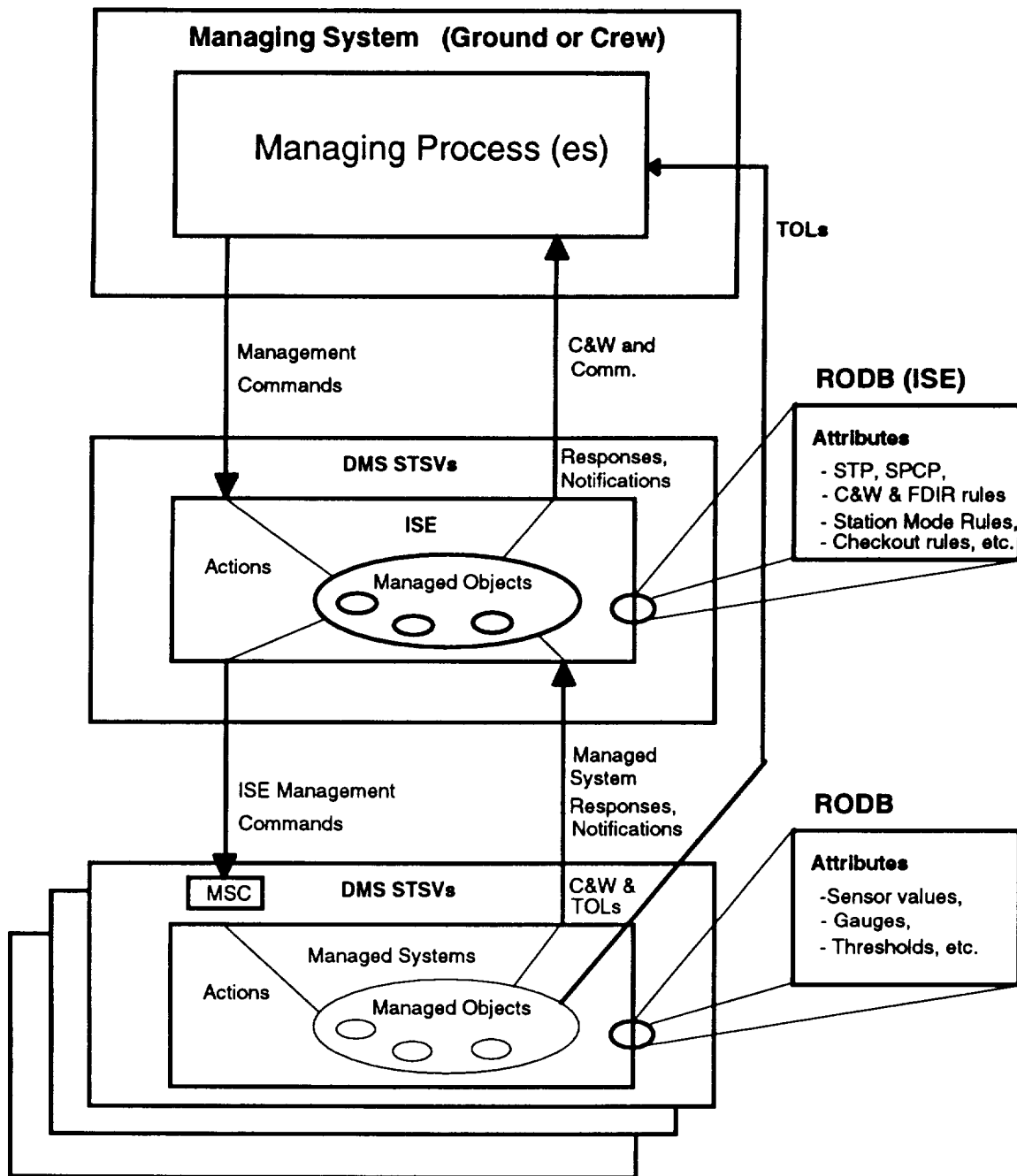


Figure 2. The ISE as a Supporting Agent to the Crew and Ground Controllers.

3.3 ISE Integrating and Coordinating Functions

The ISE in both of its roles as a Tier 1 peer and as a Tier 1 supporting agent provides for the integration and coordination of station-wide functions. The integration and coordination functions follow defined rules of behaviour to issue commands to *Freedom's* system, element, and payload application software. These defined rules of behaviour are established and enforced by the station modes described in section 2.

The ISE integrates station operations via the station and system modes by executing commands that control the systems, elements, and payloads operations both within a given mode and among the station modes. Each mode, system, element, and payload has a finite set of enable or inhibit commands. Each mode, system, element, and payload has sequences of commands that perform according to the OSTP, FDIR rules, C&W rules, test and checkout rules, and system initialization rules.

The ISE also integrates the station by enforcing resource and environmental constraints defined for the space station by considering the limits of each system, element, and payload in each station mode and for each transition among the defined modes. For example, resource constraints are limits on the combined power systems, the thermal systems, the guidance systems, and the life support control systems. Environmental constraints are those such as operations in or near the South Atlantic Anomaly, or operations constrained to a microgravity or reboost environment.

Figure 3 provides a block diagram of a possible architecture for the ISE functional applications and the relationship of those applications to the DMS, the RODB/IODB, and the managed systems, elements, and payloads. As seen from figure 3, the architecture partitions the ISE executive application into four functional areas: station mode manager, system controller, station event manager, and the operations plan manager. Each of these proposed functional areas of the ISE are discussed in detail below.

The suggested ISE architecture approach requires substantial support from the DMS software. Some of the support needed by ISE is currently not part of the DMS design. Therefore, the proposed ISE architecture also includes a set of management support objects recommended for inclusion into the DMS functional suite. These objects are the Command Sequencer and the Command Discriminator. Each of the four subfunctions of the proposed ISE architecture makes heavy use of the Command Sequencer and the Command Discriminator objects. The functions provided by these objects and their important relationship to both ISE and DMS are briefly described in this section. A detailed description of these objects is given in appendix E.

In addition, the ISE architecture assumes that all managed applications within the station's systems, elements, and payloads use a common set of object attributes and relationship attributes. This commonality is essential to minimize confusion in the design of the interfaces among ISE and the managed applications. The commonality will also aid in reducing the amount of code to be developed to support the management function and will also help to reduce processor loading servicing unnecessarily unique interfaces. A detailed proposal for the common object and relationship attributes is presented in appendix F.

Both the suggested augmentations to DMS and the common object and relationship attributes have been designed after the SMI ISO functions described in ISO/IEC 10164. As such, they provide a consistent set of definitions that comply with the basic ISO reference model for the management of open systems.

3.3.1 Station Mode Manager

A primary function of the ISE is to implement and enforce the *Freedom* station modes in support of the other TIER 1 agents and when the ISE is performing its role as a TIER 1 peer.

The station Mode Manager would implement and enforce the station modes for the ISE using integrated mode transition procedures. These procedures contain the sequence of commands required to establish the conditions, configurations, and environment needed to perform station mode transitions. Each system mode transition procedure performs a set of tasks that control the DMS RODB objects to make the mode transition. After the mode transition has been implemented, the transition procedures perform post-condition checks and report the mode status to the crew and ground.

The station Mode Manager implements the station mode transition procedures as a result of commands received from either the crew, from authorized flight controllers in the SSCC, or as scheduled in the OSTP. The ISE can also implement a mode transition as part or a predefined response to a Caution and Warning (C&W) notification or an FDIR notification. The transition commands would be received from the ISE's Message Discrimination function.

Upon receipt of a transition command, the station Mode Manager selects the proper station mode transition procedure and sends an initialize action command to instantiate a Command Sequencer object for the transition process. The Command Sequencer, using the ISE selected transition procedure file list, performs the predefined set of precondition checks, and then executes the command sequences to make the mode transition. In so doing, the station

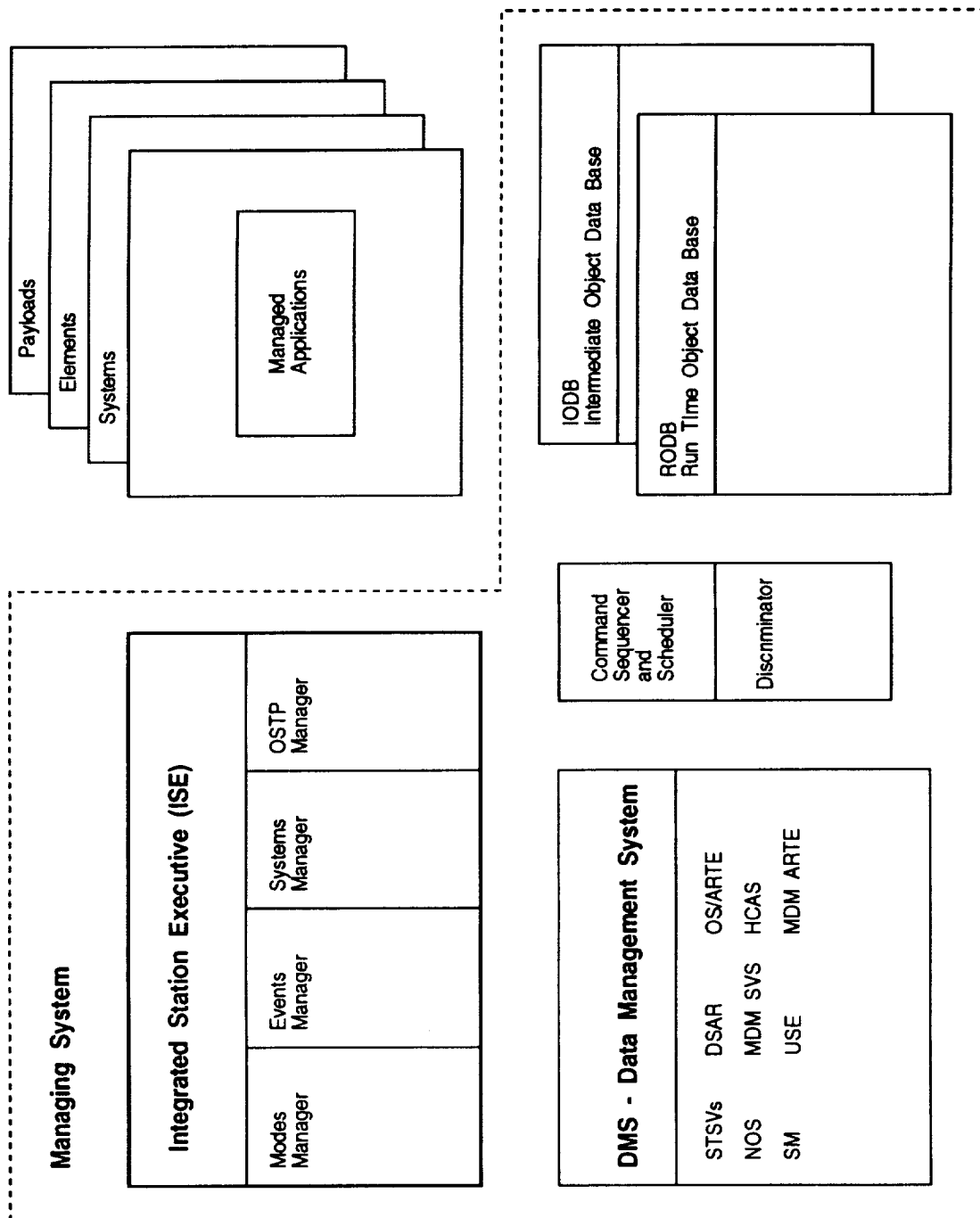


Figure 3. ISE Functional Architecture Diagram

Mode Manager, using the Command Sequencer, establishes the necessary station-wide setting of interlocks and constraints that are defined in the new mode and executes commands to systems and elements to bring them into modes compatible with the station mode. The ISE controlled command sequence would then perform a set of post-condition checks to verify the success of the mode transition and notify the crew and SSCC of the new station mode.

Aside: It is noted here that the design for the Command Sequencer object class as described in Appendix E allows for a multiple number of command procedure files to be linked to the sequencer at the time of initialization. This means that the pre-condition, mode transition, and post-condition checks could each be separate command sequences, each separately modifiable and controllable by the crew or ground. It also means that standard pre-condition and post-condition checks could be established for all integrated mode transition procedures and be frozen, thus providing additional assurance of deterministic behaviour for the station as the mode transition procedures go through stages of modification.

The command inhibits and interlocks that were established by the station Mode Manager during the mode transition process enforce the station mode. This function will not, however, preclude the crew or the SSCC from controlling operations within a given station mode. The station Mode Manager will notify the crew and the SSCC when a command is attempted that is incompatible with the current station mode. The station Mode Manager will allow the crew or authorized flight controllers to override any constraints within the current station mode, and it will allow the crew or the SSCC to respond to emergency and contingency conditions, regardless of the station mode.

3.3.2 ISE Event Manager

The ISE Event Manager subfunction carries out those routines necessary for ISE to respond to station-wide events. The Event Manager includes event pattern recognition software allowing it to compare current station-wide events with ground supplied event data and supply a response. Station-wide events could include one or more of the following:

- C&W messages processed by the DMS STSVs
- System application generated FDIR notifications
- Resource availability conflicts with OSTP data
- Environmental resource conflicts or events
- Manually input events (from crew or ground)

The ISE Event Manager, using the event discriminator object discussed below, compares the current station state with the event combination data. This event combination data is pre-

defined information in the form of tables or files, generated and uplinked by the ground, and is modifiable through established data configuration management procedures.

The event comparison function of the ISE Event Manager, using the event discriminator, is seen to be an interrupt-driven process rather than a cyclic process. Whenever a C&W message, an FDIR notification or manually input event is reported to the ISE, the current station state information is compared with the combination data for a match. The first event type reaching the ISE would be used as a key to search the event combination data.

For example, should a C&W message reach the ISE, the Event Manager would examine the event combination data for that C&W message type. If the search is successful, the ISE event manager is then given the combination of other events, which if found to also exist, would cause the Event Manager to process a response. This response, identified with the event combination data, could be a process to generate a new C&W message thus providing the ISE C&W synthesis function, or it could be a process that executes a failure reconfiguration thus providing that ISE required function.

Other event data examined by the ISE Event Manager would be derived from the execution of the OSTP. Resource conflicts, as determined by the OSTP execution process (see 3.3.4 below) by comparing available vs. planned resources, could also be an event type in the event combination data. The ISE OSTP manager would notify the ISE Event Manager of the resource conflict event. The event combination data would be searched for the event, and if found, the associated response would be executed. This same type of process would be used to process environment parameter violations.

The Event Manager subfunction is a data driven concept. The event combination data along with the predefined responses are prepared on the ground and uplinked to the ISE. Any command sequences required for the response to a detected event would be logically linked to the event combination, but could be uplinked as a separate command sequence file. Both the crew and ground would be permitted to update or modify the event combination data, adding new event combinations and refining or deleting existing sets. All such modifications would be done under strict configuration control to maintain the determinate nature of the flight software.

3.3.3 ISE Systems Manager

The ISE has the Tier I responsibility of managing and controlling all application flight software on-board. This includes application software in all systems, elements, and payloads. The ISE Systems Manager subfunction provides the following capabilities with respect to each on-board application:

System Initialization and Activation

Initialization - ISE connects power to DMS hardware on an avionics string. ISE then notifies DMS SM of the ORU availability. DMS SM will then load the system software and initialize applications.

Activation - ISE connects power to a system's ORUs. It then notifies the system of the ORU availability. The ISE activates command inhibits, enables and interlocks.

Finally, the ISE commands the system to a predefined operational state.

System Shutdown and Deactivation

Shutdown - ISE disconnects power from DMS hardware components on an avionics string. ISE then notifies DMS of the ORU non-availability.

Deactivation - ISE disconnects power from a system's ORUs on an avionics string and send commands to the system application to establish a non-operational state.

System Moding

The system moding subfunction transitions system and element application software from one system state to another. The target system or element mode is checked for compatibility with the active station mode. ISE will also verify a pre-defined set of initial conditions before commanding the system or element mode transition.

System Operational Reconfiguration

The ISE system operational reconfiguration subfunction configures system and element application hardware and software to provide redundant equipment for time critical and safety critical station operations.

System Test and Checkout

The system test and checkout subfunction of ISE system control configures systems and elements for system testing and checkout. The ISE will connect power to the system and element ORUs in the test configuration and notify the systems and elements of the ORU availability. The ISE will then command the system into a predefined test mode by establishing the necessary command enables, inhibits, and interlocks. The ISE can also support the system test and checkout by commanding predefined system test command sequences containing system and element BIT and BITE activation.

It is noted that not all of these capabilities are required or even necessary for all of the on-board applications. Consequently, the ISE functions available vary with each application.

In some cases, notably the JEM and APM, the required ISE system control capabilities are the subject of international negotiations. The requirements for ISE as they are currently understood are listed below in table 1. Exceptions to the table entries are discussed in the table notes.

The ISE system control function is derived from the ISE capability to control the execution of command sequences. This capability is achieved through the use of the standardized Command Sequencer objects described briefly in paragraph 3.3.5 and in detail in appendix E. Each of the ISE System Control subfunctions, such as system initialization, invokes the execution of a Command Sequencer that performs the required functions. Most of these system command sequences are expected to be relatively static in nature. As the station configuration of systems and elements changes with time and mission builds, the system control sequences will also require modification. With this implementation strategy, a significant portion of the ISE system control function can be frozen. System interface code and the user interface code, for example would not be expected to change. The individual command sequences applicable to each of the individual systems and elements which will change over time, can be modified, tested, and implemented without disturbing the bulk of the ISE system control application code.

The command sequences that ISE executes in order to control the various station systems, elements, and payloads will contain commands that perform the following types of functions:

- Power on/off system ORUs
- Notify systems of ORU availability
- Switch scan lists used by DMS to create downlink telemetry
- Transition between system and subsystem modes
- Activate command inhibits and command enables
- Apply and remove interlocks
- Switch MDM Input/Output Data Base (IODB) scan lists for systems located in MDMs
- Affect system operations as pre-approved

Not all of the above command types will be applicable to systems, elements and payloads under ISE control. Other applications will have unique functions for ISE to manage. In either case, ISE will make use of the standard Command Sequencer interface proposed as a DMS capability in order to control these functions.

The applications identified for which ISE has control requirements are listed in table 1.

Table 1. ISE System Control Functions versus *Freedom* Systems and Elements

	Initialize/ Activate	Shutdown/ Deactivate	Mode	Operational Reconfig.	Test and Checkout	Operational Control
DMS	X	X		X		
C&TS	X	X	X	X	X	
EPS			X	X	X	
ECLSS	X	X			X	
GN&C/P	X	X	X	X	X	
ETCS	X	X	X			
ITCS	X	X				
CHeCS	X	X	X		X	
MSC	X	X	X		X	
MTE	X	X	X		X	
EVAS	X	X	X		X	
RJ	X	X	X		X	
SDMC	X	X	X			
ULC/PMAS	X	X	X			
JEM	x1	x1	X	x1		X
APM	x2	x2	X	x2		
Racks	x3	x3		x3		
SEP				x4		x4
PES				x5		x5

NOTES:

- x1: JEM requirements for ISE identified to date do not include activation, deactivation, or test and checkout. They do include control of secondary power which is included under operational reconfiguration.
- x2: APM requirements for ISE identified to date do not include activation, deactivation, or test and checkout. They do include control of secondary power which is included under operational reconfiguration.
- x3: Rack control requirements for ISE identified to date do not include activation, deactivation, mode control, test and checkout, or operational control. They do include sending commands to SEPS, TCS, and ECLSS which is included under initialization and shutdown.
- x4: Secondary power control for ISE consists of operational control of secondary power distribution, and reconfiguration of secondary distribution under failure conditions. No initialization, shutdown, or moding control requirements have been identified.
- x5: Payload control consists of controlling secondary power distribution to payloads and issuing operationally hazardous commands under special conditions through the Payload Executive Software.

3.3.4 ISE OSTP Manager

The ISE is responsible for the control and execution of the On-Board Short Term Plan (OSTP). The OSTP provides the necessary information that allows the ISE to execute coordinated payload and core system activities. The ISE provides the capability to the crew and ground to modify the OSTP. The ISE also provides the crew with an assessment capability for all or a selected part of the unexecuted portion of the OSTP. This assessment will identify resource consumption conflicts and environmental privilege conflicts. The ISE makes all of this data available for display to the crew and for downlink to the SSCC.

For the purposes of this discussion of OSTP management, the following definitions are used. It is noted that definitions for these terms have not been established by NASA, consequently the usage given is submitted as a baseline for NASA approval.

- Timeline: A time ordered set of one or more station, system, element, or payload activities. (note: more than one activity may be active at the same time)
- Activity: A time ordered set of one or more station, system, element, or payload procedures. (note: more than one procedure may be active at the same time)
- Procedure: An ordered series of flight software command statements linked by Boolean and branching logic expressions.

3.3.4.1 Timeline Execution

The ISE uses the time data in the OSTP to execute activities at scheduled times. The ISE reads the OSTP master timeline and initializes Command Sequencers for each procedure within each scheduled activity on the timeline. The ISE will also establish any necessary scheduler relationships to control the times when the Command Sequencers are to activate the subject command procedures. The ISE allows the crew or ground to terminate, suspend, or resume the execution of the timeline. Upon command, the ISE will allow new versions or revisions of the timeline to be incorporated into the OSTP and will permit the direct execution of an activity without requiring modification to the timeline. As necessary, the ISE will prompt the crew for selected responses as part of the execution of the timeline. The ISE will compute projected start and finish times for activities on the OSTP timeline.

3.3.4.2 Activity Execution

The ISE, in executing an activity from the OSTP, initializes a Command Sequencer with the procedure identifier (file name) for each procedure contained in the activity. In addition, the

ISE will initialize the Command Sequencer with pre-condition and post-condition check procedure identifiers as contained in the OSTP. The ISE, through its control of the Command Sequencer, will allow the crew or ground to select the execution level of the activity (manual, single step, or automatic execution). Once the execution of the activity has begun, the Command Sequencer allows the ISE to terminate, suspend or resume the execution process.

The pre- and post-condition procedures initiated by the ISE in each Command Sequencer allows that Command Sequencer to examine activity pre- and post-conditions with interlock statuses, facility or equipment statuses, crew and station statuses, as well as the status of other activities on the timeline.

Through the attribute change command to the Command Sequencer, the ISE is able to perform such actions as suspending the activity execution for a specified amount of time or to change the execution step in the command sequence. Many other options are available. For more detail the reader is referred to appendix E.

The ISE also provides the capability to edit the OSTP and to assess unexecuted segments of the OSTP timeline. The ISE timeline editing capabilities include the ability to add, modify or delete activities on the OSTP timeline. Modification and deletion of activities is limited to those not already being executed. In addition, the ISE can edit the duration of any activity on the timeline as well as change its scheduled start time.

The ISE assessment capability allows the crew to examine any unexecuted portion of the timeline (or timeline edits) for conflicts between projected resource availability and projected consumption. The ISE will also examine the unexecuted portion of the timeline (or timeline edits) for conflicts between projected environmental parameter disturbances and privileges with the projected needs of planned activities.

3.3.5 Command Sequencer and Scheduler

The ISE requires a basic command sequencing and scheduling capability. This concept is basic to the requirement of performing any command function either as a Tier 1 peer or as a Tier 1 support agent. It is proposed that the ability to schedule and execute a sequence of commands or actions to affect the behaviour of all managed objects in the RODB and IODB be added to the functions provided by the DMS. The Command Sequencer proposed would include the capability to modify the timing of the execution of the command sequence and the logical branching of the command sequence. The timing of individual commands would be invoked through the scheduler function. The logical branching within a command sequence would be invoked by the sequencer monitoring expected and predefined object notifications invoked by an object action command.

It is noted that neither the Command Sequencer nor the scheduler are independently called out by the high level ISE design documentation¹⁰. Their need, however, is implied and in fact necessary for the implementation of many of the ISE and *Freedom* capabilities. Several requirement descriptions of the ISE and DMS functions indicate that the ISE has the capability to implement sequences of commands. Further, the ISE documentation implies that it has the capability to sequence its functions (OSTP implementation, TOL generation, etc.), and discriminate timetags or other control functions based on timing information.

It is suggested that the scheduling function could use the DMS distributed timing signals to set and maintain a local timing reference. The scheduling function could then perform a comparison of its local timing reference to timelines and sequences of scheduled procedures and execute commands as scheduled. It is further noted that both the sequencer and the scheduler functions could be used by many other systems, elements, and payloads. As such, they should be considered as candidates for inclusion into the suite of CSCIs provided by the DMS to the entire station.

As a result of NASA's interest in the need for a remote commanding capability in an open system's environment, MITRE has developed a proposal to the American National Standards Institute's (ANSI) open systems interconnection management committee X3T5 that a scheduler/sequencer be included into the SMI ISO standard functions set. The committee has accepted this recommendation, and a draft standard is being prepared for international consideration.

A detailed description of the Command Sequencer, as it has been proposed to the ANSI committee and as it could be implemented to support the ISE, is given in appendix E.

3.3.6 Event Discriminator

The Event Discriminator is proposed as a standard capability to be provided as a DMS service to perform the function of using predefined conditions to filter messages and object notifications to provide standard predefined responses.

The ISE C&W Synthesis capability would use this Event Discriminator to filter C&W messages and object notifications such as FDIR responses, to recognize predefined combinations in the notifications, and to generate new C&W messages. This use of the Event Discriminator allows the ISE to detect and report to the crew and ground predefined

¹⁰ This statement has been overcome by events in that later documentation included the need for either UIL/ILE or TIMELINER.

patterns of system, element, and payload faults and station-wide problems that might otherwise go undetected.

The Event Discriminator could be similarly used to provide the ISE Failure Reconfiguration function. A detailed description of the event discriminator as it could be implemented to support the ISE is given in appendix E.

SECTION 4

ISE SUPPORTING FUNCTIONS

This section describes the supporting functions that the proposed ISE architecture needs to meet its stated requirements. Many of these functions are provided by a combination of the DMS STSVs and the use of the RODB and IODB. The supporting functions provide the ISE its required management capability over the station's system, element, and payload applications. The application management support functions needed by ISE and discussed in this section include:

- Object management functions
- State management functions
- Attributes and objects for representing relationships
- Alarm reporting function
- Event reporting function
- Objects and attributes of access control

Appendix B, *Open Systems Management Tutorial*, describes the abstract model for OSI management. The international standard ISO/IEC 10040, *The System Management Overview*, provides a standardized description of the OSI abstract model and how it relates to aspects of management organization, management information, management functions, and communication services. The reader may wish to read appendix B and possibly the OSI *System Management Overview* before reading section 4.

In addition, the OSI system management standards also describe models for the management of functions that are currently assigned to the DMS, but which are of great interest to ISE and all other on-board applications. Because the OSI standard models for these functions address so many questions being addressed by SSFP developers, they are discussed in detail in the appendices. The standard functions discussed are listed below.

- Log control function
- Summarization function for telemetry selection and control
- Testing function to manage on board test and check-out
- Scheduling function to support on-board operations and commanding

For those readers who may not wish to read section 4, the authors suggest skipping to section 5 - Findings, Recommendations, Trade-offs, and Risks. After reading section 5, you may

wish to read appendix B and those parts of section 4 that relate to the recommendations and risk assessments.

4.1 Object Management Function

This section of the document describes a management function that may be used by all other application processes in the Space Station *Freedom* Program.

The Object Management function is proposed to meet the requirements of the SSFP Tier 1 to examine and change attributes of the set of managed objects that form the RODB. In addition, SSFP Tier 1 needs the ability to initialize modules, deactivate modules, and change attribute values of the managed objects that form the RODB and IODB. Tier 1 also needs to be notified of changes occurring in the configuration of *Freedom*.

A standard object management function that provides these basic needs would form a part of a systematic and flexible command structure. The following sections include a description of the object management function as standardized by ISO/IEC. This object management function meets all of the requirements of the Tier 1 components. Section 5 of this document includes findings, recommendations, trades, and risks associated with this design of a standard object management function.

4.1.1 Object Management Model

Each *Freedom* object is subject to management¹¹. The objects and their attributes are to be defined in accordance with appendix D, the Flight Software Data and Object Standard, of the DMS Architecture Control Document (ACD), (NASA, 1991 [SSP 30261]). (These data standards refer to an applicable document, the *Structure of Management Information* (SMI), ISO/IEC 10165.) SMI part 2 includes the support objects, attributes, and notifications described in the international standard, ISO/IEC 10164. ISO/IEC 10164 assumes that management information is defined according to the standard rules of SMI. (The SMI standard is the *dictionary* that lists the support objects, attributes, notifications, and behaviours. It shows how to spell in Abstract Syntax Notation One (ASN.1) basic encoding rules for the bits in the fields of communication concerning the standardized support objects.) The object management function is specified by the International Organization for Standardization (ISO) in the document, *Information Processing Systems - Open System Interconnection - System Management - Part 1: Object Management Function* (ISO, 1991

¹¹ Management: Management of the *Freedom* objects is the commanding and monitoring of *Freedom's* systems, elements, or payloads. The RODB and IODB contains management views of the attributes of all the managed objects.

[10164-1]). This standard provides complete detail on the object management function and consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the *Common Management Information Services (CMIS)*, (ISO, 1989 [9595]), and the *Open System Management Overview* (ISO, 1991 [10040]).

In the concept of the management of *Freedom* objects, the objects are transitioned to the initialized state (managed object created¹²) and transitioned to the dormant state (managed object deleted¹³), and the values of the attributes of the objects can be changed in one or more of the following three ways:

- Local management of the resources: Object attributes can be changed through the management of the configuration (reading and writing of attributes and commanding of actions) of the processes in *Freedom's* systems, elements, and payloads¹⁴ which are outside of the scope of the OSI standards.
- Management of the OSI communication services provided: Object attributes can be changed through the management of the Network Operating System (NOS) that implements the ISO (N)-layer operation and the management described in the interface control documents (ICDs) for those OSI (N)-layers agreed upon by the SSFP (See appendix G, DMS Communications Protocol Profiles, of the DMS ACD, (NASA, 1991 [SSP 30261]).)

¹² Creation: In the ISO/OSI management standards, the creation of objects means the invoking or loading of the software modules. The ISO use of the word *create* is similar to the concept of initialization in the DMS ACD. An OSI management creation does NOT imply any God-like behaviour.

¹³ Deletion: Deleting managed objects in the ISO standard is the NASA concept of transitioning to the dormant state. In the ISO standards, deletion does NOT imply the destruction of the managed object.

¹⁴ Systems, elements, and payloads: The systems, elements, and payloads are managed objects that contain other managed objects. The states of the system, element, and payload managed objects follow the same definitions that apply to the states of their contained objects. The collection of the state attributes values of the contained objects provide a detailed view of the states of the contained objects in the system, element, or payload.

- Management of the services that meet the ISO standards: Object attributes can be changed through the management of the DMS STSVs as they relate to the management of the ISE, and the on-board systems, elements, and payloads.

The international standard, ISO/IEC 10164-1, Clause 7.1, specifies these three ways of managing OSI managed objects and is, therefore, in agreement with the management scheme used in the SSFP design.

The international standard, ISO/IEC 10164-1, specifies and describes services for reporting:

- Initialization or the transition to the dormant state of managed objects
- Changes to attribute values of managed objects

The international standard, ISO/IEC 10164-1, specifies object management for commanding as listed below. The *Freedom* objects use DMS STSVs for the same configuration management functions.

- Transitioning to the initialized state (managed object creation)
- Transitioning to the dormant state (managed object deletion)
- Invoking a predefined specific behaviour of the managed objects (Actions¹⁵)
- Writing attribute values (commands that replace a value, remove a value, or replace with a default value)
- Reading attribute values (commands that get a value)
- Reporting notifications (the normal responses of the managed objects as events occur) as event reports

The object management standard maps these functions onto the underlying ISO communications services. Specifically, it maps these functions to the ISO CMIS IS, (ISO, 1989 [9595]).

¹⁵ Actions: Actions are the commands and/or services that a managed object and its application software can perform. The actions define the possible messages to which the object will respond. Action commands result in object behaviour.

In addition to using the ISO CMIS standard, DMS STSVs are mapped to the underlying communications services of the Consultative Committee for Space Data System (CCSDS) standard. DMS STSVs uses CMIS for communication to Tier 1 components and for communications among the *Freedom* systems, elements, and payloads. DMS services use the CCSDS underlying communication services for sending summary notification (TOLs). The reader is referred to appendix G of the DMS ACD for details.

4.1.2 Object Management Generic Notification Definitions

This section describes the set of four notifications and their applicable parameters¹⁶ and semantics specified by the international standard ISO/IEC 10164-1.

4.1.2.1 Object Initialization Notification

If the class of objects requires reporting of the transition to the initialized state, then that managed object class imports a common object **creation notification** type. (This notification type could be supplied by a mapping of the DMS C&W standard service.)

The notification should include the following mandatory or optional parameters¹⁷:

- The mandatory **object creation notification** parameter that indicates the type of notification.
- The optional parameter set consisting of a **source indicator** and **additional information**.

¹⁶ Parameter of a notification: A parameter of a notification is the reported bit field that is to be filled with an attribute value. The coding of the attribute value in the parameter is to follow the standardized ASN.1 transfer syntax as specified in the ISO/IEC IS 10165-2.

¹⁷ Optional parameters of notifications: Optional parameters of notifications are fields of the report that may be included in the notification services if the user chooses. Mandatory parameters must be in the notifications.

- The optional **source indicator** has one of the following types:
 - **InternalResource**--the notification was generated in response to an initialization command through the internal operation of the objects (i.e., the systems, elements, or payloads).
 - **LocalOpenSystem**--the notification was generated in response to an initialization command applied across the managed object boundary but from within the managed object (i.e., The initialization was commanded by the ISE or the crew.).
 - **RemoteOpenSystem**--the notification was generated in response to an initialization command initiated from a remote manager (i.e., the SSCC, POIC, or any other system [an internal partner's control center]).
 - **Unknown**--it was not possible to determine the source of the operation.
- The **additional information** is provided to convey specific object class information associated with the **create notification**. (For example, this information could include the source address of the command.)

4.1.2.2 Object Transition to the Dormant State Notification

If the class of *Freedom* objects requires reporting of the transition to the dormant state, then that managed object class imports a common object **deletion notification type**. (This notification type could be supplied by a mapping of the DMS C&W standard service.)

The international standard specifies that the **deletion notification** should include the following mandatory or optional parameters:

- The mandatory **object deletion notification** parameter that indicates the type of notification.
 - The optional **parameter set** consisting of a **source indicator** and **additional deletion data**.
- The optional **source indicator** has the one of the following types:
 - **InternalResource**--the notification was generated in response to a dormant command through the internal operation of the managed objects.

- **LocalOpenSystem**--the notification was generated in response to a dormant command applied across the managed object boundary but from within the managed system.
 - **RemoteOpenSystem**--the notification was generated in response to a dormant command initiated from a remote managing system.
 - **Unknown**--it was not possible to determine the source of the operation.
- The **additional deletion data** is provided to convey specific object class information associated with the **delete notification**.

4.1.2.3 Attribute Value Change Notification

If the class of *Freedom* objects requires the capability to report attribute values changed, then that managed object class imports a common **attribute value** notification type. (This notification type could be supplied by a mapping of the DMS attribute change notification service.)

Depending upon the omission of attribute values in the telemetry object lists, it may be important to notify the commanding Tier 1 entity of an attribute value that was commanded to change. Examples could be as follows:

- Enabling¹⁸ access of one or more commands to a system, element, or payload
- Inhibiting¹⁹ access of one or more commands to a system, element, or payload
- Replacing of the value of one or more attributes of a managed object
- Changing of the value of one or more attributes to their default value(s)

¹⁸ Command enabling: A command enable provides the access to a device or an application to execute commands. The removal of command inhibits provides command enabling.

¹⁹ Command inhibiting: A command inhibit temporarily removes from a device or software application the ability to send, receive, or execute a command. Inhibiting the ability to send provides access control by preventing the sending of the command. Inhibiting the ability to receive provides access control by a decision function that blocks the command from being delivered to the managed object. Inhibiting execution of a command prevents the behaviour of the managed object (i.e. prevents the behaviour of the system, element, or payload by limiting the activity of the application).

Note this object management notification type should **NOT** be used for conveying attribute information changes that have specific notification types defined as a part of the object definitions provided to meet the flight software and object standard. Nor should it be used to convey attribute information where TOLs have been defined.

4.1.2.4 Notification Parameters

The standard specifies that all notifications should include the following mandatory or optional parameters:

- The mandatory **attribute change notification** parameter that indicates the type of notification.
- The attribute change parameter set consisting of the mandatory **attribute change definition** parameter, optionally followed by the Additional Info Parameter.

The mandatory **attribute change definition** parameter is a set of sequences of the following four parameters: **Attribute ID**, **Old Attribute Value**, **New Attribute Value**, and **Source Indication**. Each sequence of the four parameters indicates a single attribute change. The mandatory **attribute change definition parameter** set contains two mandatory parameters and two optional parameters. The elements of the **attribute change definition parameter** set is as follows:

- The **attribute ID** identifies the attribute whose value change is being reported and is a mandatory parameter in the set.
- The **old value** of the attribute is an optional parameter.
- The **new attribute value** identifies the current attribute value and is a mandatory parameter in the set.
- The optional **source indicator** indicates the source of the operation that commanded the generation of the attribute change and resulted in the generation of this notification type. The optional source indicator has one of the following types:
 - **InternalResource**--the notification was generated in response to an attribute change command through the internal operation of the managed systems.

- **LocalOpenSystem**--the notification was generated in response to an attribute change command applied across the managed object boundary but from within the managed system.
 - **RemoteOpenSystem**--the notification was generated in response to an attribute change command initiated from a remote managing system.
 - **Unknown**--it was not possible to determine the source of the operation.
- The optional **additional info** data is provided to convey object class specific information associated with the reason for the **attribute value change** notification.

4.1.3 Object Management Service Definitions

The object management function will be provided by the detailed design of the DMS. The DMS should have standard services to provide the services listed in the object management model and object management notification sections. Examples of how DMS could map these services to CMIS are provided in the ISO/IEC IS 10164-1. The design of the DMS does not have to comply with that standard, but the capability of DMS will require the functions of the standard.

4.1.4 Object Management Protocol and Abstract Syntax Definitions

The Flight Software Data and Object Standards, appendix D of the DMS ACD, calls for the applicable document ISO/IEC 10165. This ISO/IEC standard, ISO/IEC 10165-2, defines abstract syntax for the following notification types and their parameters:

- objectCreation
- objectDeletion
- attributeValueChange

4.2 State Management Function

This section of the document describes a management function that may be used by all other application processes in the Space Station *Freedom* Program.

The State Management function is proposed to meet the requirements of the SSFP Tier 1 to monitor and control the activities of the managed objects that form the system, elements and

payloads. In addition, SSFP Tier 1 needs a consistent set of definitions related to management of the states of *Freedom's* systems, elements, and payloads. Tier 1 also needs to be notified of these states of the configuration of *Freedom*.

In the concept of the management of *Freedom* managed objects, Tier 1 needs the ability to examine and be notified of changes in state; to monitor overall operability and usage of systems, elements, and payloads, in a consistent manner; and to control the general availability of specific systems, elements, and payloads as a function of station modes. In some cases, the systems, elements, and payloads will be subject to both command constraints²⁰ and managed object (system, element, or payload) resource constraints²¹ that limit the availability and utilization of the objects. The attributes values related to the resource constraints attributes are used to check and make the transitions among the states of the objects.

The state of a managed object represents the instantaneous conditions of availability and operability of the object and its associated systems, elements, and payloads from the point of view of Tier 1 management. Different classes of managed objects have a variety of state attributes that express and control aspects of the operation of their associated resources. Nevertheless, the state can be common to a large number of systems, elements, and payloads. For this reason, it is desirable to standardize the state management function. The standardization of state management is to control the general availability of the systems, elements, and payloads to make visible information about the general availability. If a *Freedom* managed object (system, element, or payload) is not usable, then the states indicate what kind of command (action) needs to be taken to make it usable.

A standard state management function that provides these basic needs would provide a systematic and flexible command structure. The following sections include a description of the state management function as standardized by ISO/IEC. This state management function meets the needs of the Tier 1 components. Section 5 of this document includes findings,

²⁰ Command constraints: Command constraints are sets of command inhibits to restrict and block the listed commands from reaching or affecting a managed object.

²¹ Resource constraints: Resource constraints are attribute value limits associated with managed objects that represents consumable resources. For example, most managed objects have power consumed and heat generated attributes. Depending on the operational states and power states of the managed objects and the number of managed objects (system, element, or payload), the sums of the power consumed and heat generated will determine the total power consumed and heat generated. The limits on the sums or on the individual attribute values are resource constraints.

recommendations, trades, and risks associated with this design of a standard state management function.

4.2.1 State Management Model

Each *Freedom* object is subject to state management. The objects and their attributes are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP 30261]). These data standards refer to as an applicable document, the SMI (ISO, 1991 [10165]). SMI part 1 and part 2 use the ISO functions described in the ISO/IEC 10164. iThe ISO/IEC 10164-2, *Information Processing Systems - Open System Interconnection - System Management - Part 2: State Management Function* specifies the state management function. This specification provides complete detail on the state management function and consistently defines terms that comply with the basic reference model (ISO, 1984 [7498-1]), the Open System Management Framework (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), and the Open System Management Overview (ISO, 1991 [10040]).

The international standard lists and specifies four primary factors that affect the state of managed objects (i.e., systems, elements, and payloads) with regard to its corresponding availability. Not all of these factors are applicable to every managed object. These are as follows:

- **Operability:** Whether the managed object (system, element, or payload) is physically installed and working.
- **Usage:** Whether the managed object (system, element, or payload) is actively in use at a specific instant, and if so whether or not it has spare capacity for additional users at that instant. A managed object (system, element, or payload) is "in use" when it has received one or more commands or requests for service that it has not yet completed or otherwise discharged, or when some part of its capacity has been allocated, and not yet retrieved, as a result of a previous command or service request.
- **Administration:** Permission to use or prohibit²² against using the resource imposed through management services.

²² Prohibits: The administrative permissions and prohibits are controls used to stop commands or processes while administration (the crew or the SSCC ground controllers) monitors the attributes of the managed objects and control the managed objects. The

- **Status:** Status contains more detailed information about other aspects of the state of the corresponding managed object (system, element, or payload) that may affect its operability and usage. Status states are used to support the FDIR aspects of the objects in the managed object (system, element, or payload).

The state of systems, elements, and payloads do not affect their ability to be managed.

4.2.1.1 Operational State

The operability of objects within systems, elements, and payloads is described by the operational state attribute, that has two possible values: object **disabled**, and object **enabled**. These two state attributes are described in ISO/IEC 10164-2 clause 8.2.1. Figure 4 illustrates the operational state diagram.

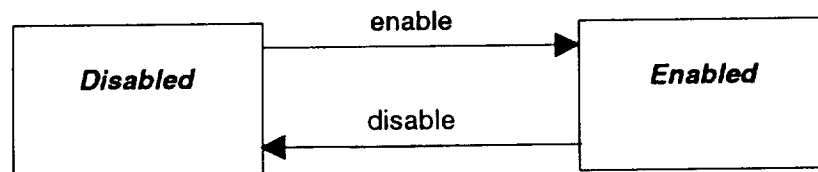


Figure 4. Operational State Model for Managed Objects

On the space station, some classes of managed objects (for example, firmware modules) exhibit only a constant enable value for the operational state. When an object within systems, elements, and payloads has no dependencies on other objects and no components that can develop visible defects, the managed object may not exhibit the **disabled** operational state. Likewise, a managed object that ceases to provide services when the managed object (system, element, or payload) becomes inoperable does not exhibit the **disabled** operational state. When a managed object (system, element, or payload) ceases to supply services, but there is still a managed object maintaining state attributes about the object within the managed object (system, element, or payload), then the operation state becomes **disabled**. For example, if ISE is tracking the operational state of a secondary power relay when the primary power distribution affecting the secondary power relay is switched off, the secondary power relay is operationally **disabled** even if the relay switch position is normally closed.

administrative permission and prohibits are effectively locks on the use of the managed resources.

It is the natural operation (behaviour) of the managed objects (systems, elements, and payloads) that cause the operational state transitions of managed objects to occur; therefore, Tier 1 cannot command a managed object to change from one operational state to another. Tier 1 or any other requester of information can only gather information about the operational state of a managed object; i.e., the operational state is a read-only attribute. (Note that if a managed object has a defined action command that results in an object behaviour that transitions the operational state, then reading the operational state attribute confirms that the action command resulted in the managed object behaviour.)

Thus, managed object specific events associated with the managed object (system, element, or payload) cause specific transitions from one operational state value to another. These events and transitions are defined as follows:

Object enable event: This event consists of action being taken to make the managed object (system, element, or payload) partially or fully operational. This event can occur only if the object of the managed object (system, element, or payload) is **disabled**. The object enable event causes a transition to the object **enabled** state.

Object disable event: This event consists of action being taken to make the managed object (system, element, or payload) totally inoperable. The object disable event causes a transition to the **disabled** operational state.

4.2.1.2 Usage State

The usage of a managed object (system, element, or payload) is described by the usage state attribute that has three possible values: **idle**, **active**, and **busy**. These states are the normal run-time states that describe the normal run-time envelope of the systems, elements, and payloads. These state attribute values are described further in ISO/IEC 10165-2, clause 8.1.1.2. Figure 5 illustrates the object usage state model.

On *Freedom*, some classes of managed objects in the systems, elements, and payloads exhibit only a subset of the possible usage state values. The managed objects that support only one user do not exhibit the **active** usage state, they are either **idle** or **busy**. The objects that have no practical limit on the number of users do not exhibit the **busy** usage state (for example the on-board local area network). The set of usage state values supported is specifically in the managed object definitions of each individual managed object.

It is the natural operation (behaviour) of the managed object (system, element, or payload) that causes usage state transitions to occur; therefore, Tier 1 commands cannot request a managed object to change from one usage state to another. Tier 1 or any other managed

object (system, element, or payload) can only gather information about the usage state of any other managed object (system, element, or payload). The usage state is a read-only attribute.

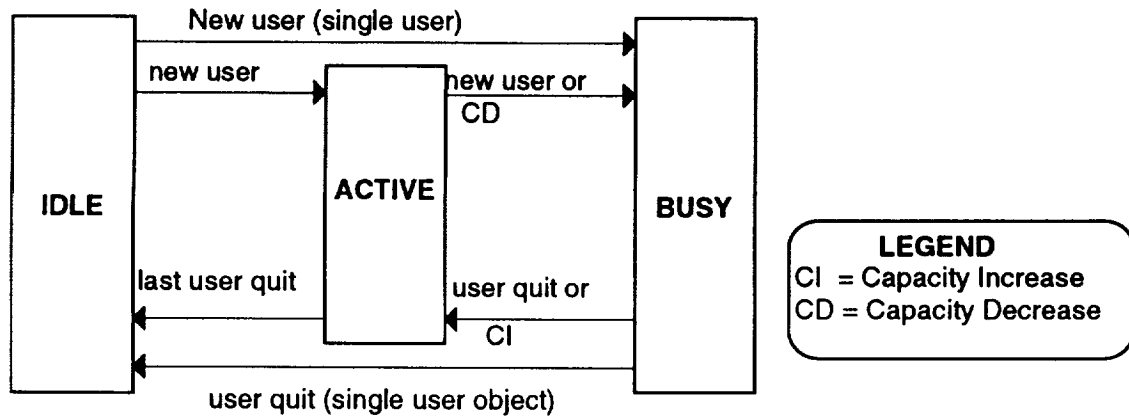


Figure 5. Usage State Model for Managed Objects

Specific reasons associated with the managed object (system, element, or payload) cause transitions from one object usage state to another. These reasons and transitions are summarized as follows:

New user transition: This consists of some user commencing to command the contained managed objects in the system, element, or payload. This transition can occur only if the contained managed object's operational state is object **enabled** and its usage state is either **idle** or **active**. The contained managed object new user causes a transition if, after the new user begins, the resource represented by the contained managed object in the system, element, or payload has:

- Sufficient capacity to provide for additional users, the usage state becomes or remains **active**. (For example managed objects such as, queues, *journalizing services*, mass storage units, recorders, C&T, Primary Power System, etc., may have spare capacity.)
- No operating capacity to spare for additional users, the usage state becomes **busy**.

Object user quits transition: This transition consists of an existing user of the managed object terminating its use. It can occur only if the managed object usage state is either **active** or **busy**. It can result from a change of operational state from object **enabled** to object

disabled. The object user quitting causes a transition if, after the user quits, the contained managed object in the system, element, or payload has:

- Existing users, the usage state becomes or remains **active**.
- No users, then the usage state become **idle**.

Capacity increase transition: This transition consists of an increase in the maximum operating capacity of the managed object. It is significant only if the managed objects usage state is **busy**. The capacity increase causes a transition to the **active** state if the managed object is in the **busy** state. Capacity increase transitions occur if attribute values representing the capacity of the managed objects increase.

Capacity decrease transition: This transition consists of a decrease in the maximum operating capacity of the managed object. It is significant only if the managed object's usage is **active**. The capacity decrease causes a transition as follows:

- If, after the transition, the managed object still has spare operating capacity, the usage state remains **active**.
- If, after the transition, the managed object has no spare capacity, the usage state becomes **busy**.
- If the managed object is in the **busy** state when a capacity decrease occurs, the managed object will continue to reside in the **busy** state until either a capacity increase transition or a user quit transition occurs.

4.2.1.3 Administrative State

The administration of managed objects operates independently of the operability and usage of managed objects and is described by the administrative state attribute, which has three values. These three values are called **locked**, **unlocked**, and **shutting down** and are described further in ISO/IEC 10164-2, clause 7.1.3. Figure 6 illustrates the administrative state model.

Some classes of managed objects exhibit only a subset of the possible administrative state values. Some systems cannot be shut down gracefully, and hence their corresponding managed objects do not exhibit the **shutting down** state. The actual subset of administrative state values supported varies from one class of managed object to another and is specified in each individual managed object definition. For example, commands from Tier 1 can request a managed object to change from one administrative state to another. Tier 1 can gather

information about the administrative state of any object of a managed object (system, element, or payload). The administrative state is a read-only attribute and its value is the result of the behaviour of the object, its use, and the commands from Tier 1.

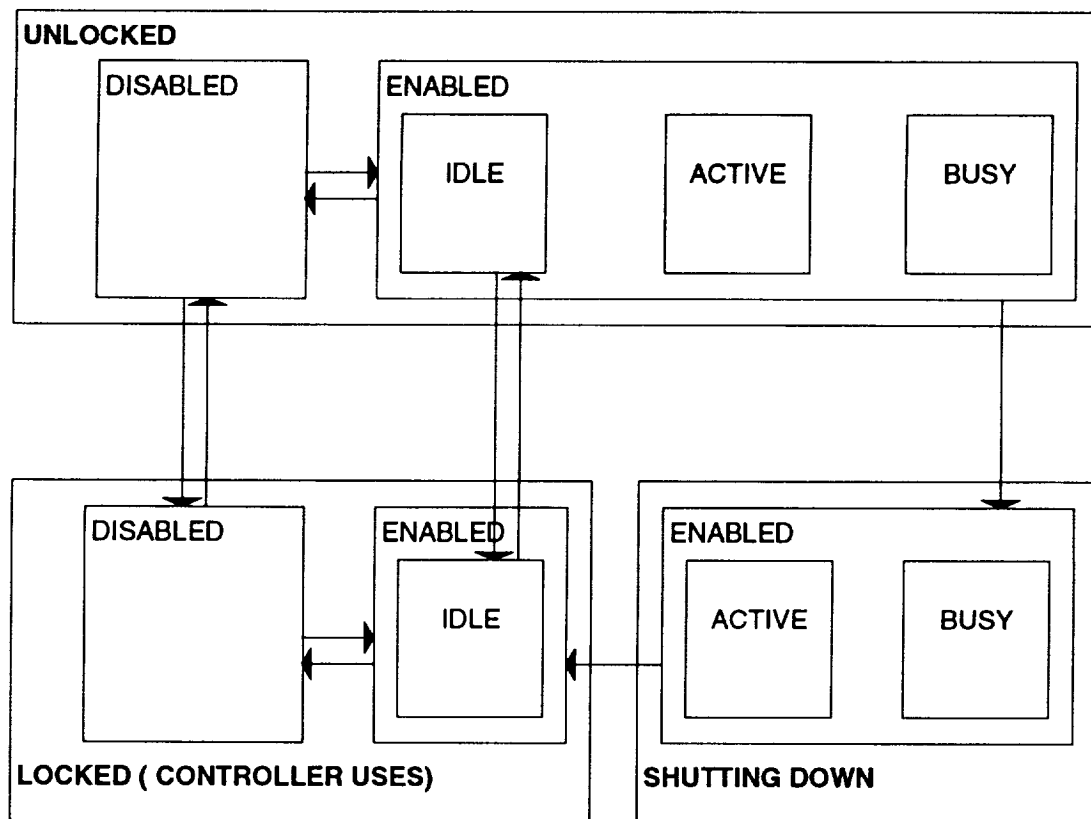


Figure 6. The Administrative State Model for Managed Objects

The specific events associated with the managed object cause specific transitions from one administrative state value to another, depending upon the original value of the administrative state, the specific event, and upon the number of users of the resource. These events and transitions are summarized in the following description of events.

Unlock event: This event consists of an operation (a sequence of commands) being performed at the managed object to unlock its process (behaviour). The event can occur only if the managed object's administrative state is **locked**²³ or **shutting down**. It causes a

²³ Locked managed objects: The administrative lock prevents user's access to the behaviour of managed objects. The administration authority continues to have access to the

transition to the **unlocked** administrative state. For example, this event would occur in the process of affecting a change in space station mode. The unlock event could occur at the completion of a station mode transition. The unlock event would allow the behaviour of the managed objects to become available to the users. Only those systems, elements, and payloads affected by the station mode would experience the unlock event.

Lock event: This event consists of an operation (a sequence of commands) being performed at the managed object to lock its corresponding process (behaviour). The lock event causes a transition to the **locked** administrative state. For example, the **locked** administrative state could be used to prevent a managed object from receiving any commands from any source other than the current space station commander. The **locked** administrative state is independent from the operational and usage states and does not prevent the operation of the managed object. Another example of the lock event would occur for all systems, elements, and payloads affected by a station mode transition requiring the complete control of those systems, elements, and payloads during the station mode transition.

Shut down event: This event consists of an operation (a sequence of commands) being performed at the managed object to shut down its process (behaviour). The shut down event can occur only if the managed object's administrative state is **unlocked**. For example, the shut down event occurs during a station mode transition that requires the deactivation of managed objects contained in the systems, elements, and payloads. It causes a state transition if, at the time of the event, the managed object contained in the managed object (system, element, or payload) has:

behaviour of the managed objects. If the current administration authority were a null set, then the locked managed object externally inhibits the managed object. (See definition of inhibits.) SSFP administrative authority consists of any authorized source. The possible administrative authorized sources include crew, the SSCC ground controllers, the POIC controllers, and international partner personnel.

Locked DMS objects: The DMS object lock prevents user's access to the replacement of RODB values. The current command authority continues to have access to the replacement of the RODB values. If the current command authority were a null set, then the locked DMS object externally inhibits the DMS object. (See definition of inhibits.) Notice that if all commands to an object are required to be inhibited, then one way to implement command inhibits would be for all commands to write an enable command attribute. This enable command attribute would need a DMS object to store the written state of the command. A DMS object lock would then prevent any unauthorized change in the enable command attribute, therefore, inhibiting the command.

- Existing users, the administrative state becomes **shutting down**
- No users, the administrative state becomes **locked**

User quits event: This event consists of an existing user of the resource terminating its use. It can occur only if the managed object administrative state is **unlocked** or **shutting down**. If the administrative state is **unlocked**, no administrative state transition occurs. If the administrative state is **shutting down**, the user quit event causes a transition if, after the event, the object in the managed object (system, element, or payload) has:

- Existing users, the administrative state remains as **shutting down**
- No users, the administrative state becomes **locked**

4.2.1.4 State Attribute

The state attribute is a group attribute that includes the combination of the operational state, the usage state, and the administrative state. The state attribute is a read-only attribute. The value depends upon the combination of the operational, usage, and administrative state attribute each managed object supports. Changes in the value of the state attribute are not conveyed by the attribute value change notification type. The values of the state attribute may be included in summarization notifications (TOLs) or they may be read by a DMS service (ACTION READ).

4.2.1.5 Status States

The status states of the station's managed objects are independent of the operability and usage states. The status states contain more information about the administrative constraints²⁴ on the object's processes (behaviour). These status state values are described further in ISO/IEC 10164-2, clause 8.1.2. The status attribute values applicable to the administrative constraints are as follows:

The **alarmStatus** attribute is a set-value and read-write. (Note: in this document, the international method of creating object class, attribute type, and attribute value names is used. To make a name, descriptive phrases are concatenated and capital letters start each word in the concatenation.) If applicable to and defined in the object class, it can have one or more of the following values:

²⁴ Administrative constraints: Administrative constraints are restrictions on the usage and availability of managed objects. For example, administrative constraints include power status, failure status, test status, preference status, etc.

- **UnderRepair:** The object is currently being repaired. When the under repair value is present, the value of the availability status attribute is **failed**. The operational state of the managed object is **disabled** or **enabled**.
- **AlarmOutstanding:** One or more FDIR alarm messages with probable cause indicating a fault has been reported for the managed object (system, element, or payload) and has not been cleared. (Note alarm messages are defined in ISO/IEC 10164-4 and are described in section 4.4 of this document.) These faults may or may not have been disabling. If the operational state is enabled, additional attributes particular to the object class, such as built-in test results, indicate the services that are affected and the nature of the fault.
- The **critical** severity level indicates immediate corrective action is required. (See section 4.4 for a discussion of OSI severity levels.)
- The **major** severity level indicates urgent corrective action is required.
- The **minor** severity level indicates corrective action should be taken to prevent serious failure.

The **proceduralStatus** attribute is set-valued and read-only. These attribute values help to determine the space station procedural state. The procedural status is the run-time envelope supporting the initialization and termination of *Freedom* managed objects. This includes the initial start-up of the station, the start-up of partial assemblies, the start-up from a shut down, the sectional start-up from a component failure, the sectional start-up from sectional upgrades of hardware or software, the recovery from power outages, and the activation of an off-line or standby unit as part of FDIR. It can have one or more of the following values, not all of which are applicable to every class of managed object:

- **InitializationRequired:** The managed object (system, element, or payload) requires initialization before it can be available for use, and this procedure has not been initiated. The manager (Tier 1) may be able to invoke such initialization through an action command. The operational state is **disabled**.
- **Initializing:** The managed object (system, element, or payload) requires initialization before it can be available for use, and this procedure has been initiated but is not yet complete. When the condition is present, the initialization required condition is absent, since initialization has already begun. The operational state is **disabled**.

- **Reporting:** The managed object (system, element, or payload) is in the process of reporting (generating a notification as part of its predefined managed object behaviour. When the condition is present, the operational state is **enabled**.
- **Terminating:** The managed object (system, element, or payload) is in the process of transiting to the dormant state. When the condition is present, the operational state is **enabled**.

The **availabilityStatus** attribute is set-valued and read-only. The status attribute supports the determination of the station checkout state, the station initialization state, the station dormant state, and the station safing state. The station checkout state is the run-time envelope supporting on-board checkout. The station dormant state is the run-time envelope supporting low power consumption. The station safing state is the run-time envelope supporting the safety of the crew. It can have one or more of the following values, not all of which are applicable to every class of managed object on-board the space station:

- **InTest:** The managed object (system, element, or payload) is undergoing a test procedure. If the administrative state is locked or shutting down, then other users are precluded from using the object and the control status attribute has the value **reservedForTest**. Tests that do not exclude the use of the object do not require the establishment of the **reservedForTest** value in the control status attribute.
- **Failed:** The managed object (system, element, or payload) has a fault that prevents the object from operating correctly. The failure has been detected by an internal check, as opposed to human speculation. The operational state is object **disabled**.
- **PowerOff:** The managed object (system, element, or payload) requires power to be applied and is not powered on. For example, a standby unit that has not failed. The operational state is **disabled**. **PowerOn** is the complement to **powerOff** except that the operational state can be either object **disabled** or object **enabled**.
- **OffLine:** The managed object (system, element, or payload) requires some switching operation (sequence of commands) to be performed to make it available for use. The switching operation may be manual or automatic or both. The operational state is object **disabled**. (The off-line attribute could help determine the station dormant mode. In combination with the **powerOff** value, the station dormant status is determined.) **OnLine** is the complement to **offLine**.

- **OffDuty:** The managed object (system, element, or payload) has been made unavailable in accordance with an on-board operating plan. Some command and control processes within the command and control structure have taken the managed object (system, element, or payload) out of service at a scheduled time. The operational state is object **disabled**. **OnDuty** is the complement to **offDuty**.
- **Dependency:** The managed object (system, element, or payload) cannot operate because some other resource on which it depends is **disabled**. For example, a device is not accessible because its controller is powered off. The operational state is object **disabled**. **NotDependent** is the complement to **dependency**.
- **Degraded:** The managed object (system, element, or payload) has degraded in service, such as in speed or operating capacity. Failure of test or an unacceptable performance measurement has established that some or all services are not functional or are degraded due to the presence of a defect, fault, or error. However, the managed object (system, element, or payload) remains available for service, either because some services are satisfactory or because degraded service is preferable to no service at all. Object specific attributes may be defined to represent further information. For example, the failure isolation and recovery services may indicate which services are not functional. The operational state is object **enabled**.
- **NotInstalled:** The managed object (system, element, or payload) is not installed or is incompletely installed. For example, a plug-in module is missing or a cable is disconnected. **Installed** is the complement to **notInstalled**. The operational state is **disabled**.
- **LogFull:** The managed object class of log is reporting a log full condition indicating that the managed object class instance is not available.

The **controlStatus** attribute is read-write and set-valued. The control status attribute is the set of attributes that support the operations for management service controls. These attributes are related to command inhibits, command constraints, command overrides, interlocks, and command procedures. It can have one or more of the following values, not all of which are applicable to every class of managed object:

- **SubjectToTest:** The managed object (system, element, or payload) is currently under test. The administrative state is **unlocked**.

- **PartOfServicesLocked:** This value indicates whether a Tier 1 inhibit has administratively restricted a particular part of a service from the users of the managed object (system, element, or payload). Examples are command constraints, outgoing message discriminators on FDIR reports or TOLs.
- **ReservedForTest:** The managed object (system, element, or payload) has been made administratively unavailable to normal users because it is undergoing a test procedure. The administrative state is object **locked**.
- **Suspended:** Service has been administratively suspended to the users of the resource. The administrative state is **locked**.

Additional state change information is a parameter of the notifications that report status attributes. Special space station attributes can be reported in this parameter. The ISO standards name the parameter: **additionalStateChangeInfo**. The SSFP **override** attribute and the SSFP **readOnly** attribute are proposed to be included in the **additionalStateChangeInfo** parameter.

- **Override:** (The override value is not a standardized attribute of ISO/IEC 10165-2.) This value would indicate that an override has been issued to the managed object (system, element, or payload). The attribute value is set depending on override events that are related to allowing the commanding of the managed object (system, element, or payload) without regard to command constraints, object resource constraints, or thresholds, or object behaviour constraints. The commanded object is forced to produce a result that is contrary to the normal logic of its software process. The override event is established through one or more of the following:
 - The administrative lock event which takes control of an object from its users and allows control from another commander.
 - The interactive control event of an otherwise automated process.
 - The commanding of an abort, of a reverse, of a change decision, or of a reconfiguration made by a software process.
 - The commanding of a reversal of some barrier to command execution, such as a command inhibit or a command constraint.
- **ReadOnly:** Tier 1 may require the suspension of the write capabilities of the DMS data objects or managed system (system, element, or payload) for a period of time. For this purpose the read only value is provided to indicate that write operations by

the managed object (system, element, or payload) are administratively prohibited on the DMS objects. This is a command inhibit that prevents DMS ACTION WRITE services on DMS objects.

The standby status attribute is used only if a back-up relationship exists. (Section 4.3 includes attribute definitions of the relationship attributes.) The standby status attribute indicates if the back-up managed object is a hot standby, a cold standby or providing service. The attribute is single-valued and read-only. The standby status attribute has the following attribute values:

- **ProvidingService:** The back-up resource is providing service and is backing up another resource. The providing service condition is mutually exclusive with the hot standby and cold standby conditions.
- **HotStandby:** The resource is not providing service, but is operating in synchrony with another resource that is to be backed-up (e.g., a computer shadowing another computer). A resource with a hot standby status will be immediately able to take over the role of the resource to be backed-up, without the need for initialization activity, and will contain the same information as the resource to be backed up. The hot standby condition is mutually exclusive with the cold standby and providing service conditions.
- **ColdStandby:** The resource is to back-up another resource, but is not synchronized with that resource. A resource with a cold standby status will not be immediately able to take over the role of a resource to be backed up and will require some initialization activity.

4.2.2 State Change Notifications

There is one ISO standardized notification type defined for reporting the change in the values of one or more of the standardized state attributes of a managed object. The reported change may result through either the internal operation of the managed on-board system (system, element, or payload) or via command operations directed to the managed on-board system (system, element, or payload). If the state attributes are included in the normal space station TOLs from the systems and elements, then the state change notification should have a discriminator so that it can be disabled. The advantage to using the standardized notification is efficiency of communication bandwidth. Using the standardized notifications removes the reporting of state variables from the TOLs. The advantage to using TOLs to report state variables is that the SSCC has periodic updates to indicate that the states have not changed. The standardized state change notification uses the parameters shown in table 2.

Table 2. Parameters for the State Change Notification

PARAMETER	Mandatory/Optional /Conditional
source Indicator	Mandatory
notification type	Mandatory
event time	Optional
list of parameters corresponding to the state attribute whose values change	Mandatory
old values of the operational state attribute	Optional
current values of the operational state attribute	Conditional
old values of the usage state attribute	Optional
current values of the usage state attribute	Conditional
old values of the administrative state attribute	Optional
current values of the administrative state attribute	Conditional
old values of the alarm status attribute	Optional
current values of the alarm status attribute	Conditional
old values of the procedural status attribute	Optional
current values of the procedural status attribute	Conditional
old values of the availability status attribute	Optional
current values of the availability status attribute	Conditional
old values of the control status attribute	Optional
current values of the control status attribute	Conditional
correlated notifications	Optional
additional text	Optional
additional state change info	Optional
current time	Optional

4.2.3 State Management Service Definitions

The state management function will be provided by the detail design of the DMS. The DMS should have standard services to provide the services listed in the state management model and state management notification sections. Examples of how DMS could map these services to CMIS are provided in the ISO/IEC IS 10164-2. The design of the DMS does not have to comply with ISO/IEC IS 10164-2, but the capability of DMS will require the functions of the standard.

4.2.4 State Management Protocol and Abstract Syntax Definitions

The Flight Software Data and Object Standards, appendix D of the DMS ACD, calls for the applicable document ISO/IEC 10165. This ISO/IEC standard, 10165-2, defines standard abstract syntax for the following state management attributes.

- additionalStateChangeInfo
- state
- operationalState
- oldOperationalState
- newOperationalState
- usageState
- oldUsageState
- newUsageState
- administrativeState
- oldAdministrativeState
- newAdministrativeState
- alarmStatus
- oldAlarmStatus
- newAlarmStatus
- proceduralStatus
- oldproceduralStatus
- newproceduralStatus
- availabilityStatus
- oldAvailabilityStatus
- newAvailabilityStatus
- controlStatus
- oldControlStatus
- newControlStatus
- standbyStatus
- oldStandbyStatus
- newStandbyStatus

The ISO/IEC IS 10164-2 also defines abstract syntax for the mapping the parameters of the state change notification to the parameters of CMIS. (See ISO/IEC IS 10164-2, clause 11.).

4.3 Attributes for Representing Relationships

This section of the document describes the relationship²⁵ attributes that may be used by application processes in the Space Station *Freedom* Program. These relationship attributes are proposed to meet the requirements of the SSFP Tier 1 to monitor and control station relationships such as back-up and backed-up objects, primary and secondary objects, station modes and system objects supporting station modes, and owner and member objects. *Freedom* has such relationships and Tier 1 needs to understand any changes in such relationships. For example, each station mode could be considered a collection of *Freedom*-managed systems (system, elements, or payloads) working to a set of station mode rules. The transitioning among these station modes would show changes in command constraints, object resource constraints, configuration state attributes (see the state management function, section 4.2), and rules governing the interactions among the systems, elements, and payloads. Also, if objects within the systems, elements, and payloads are backed-up, then when the system switches its back-up and backed-up units, Tier 1 management needs to know the relationship has changed.

Standard attributes for representing relationships that provide these basic needs would form a part of a systematic and flexible command structure. The following sections include descriptions of the attributes for representing relationship standardized by ISO/IEC. These attributes meet the needs of the Tier 1 components. Section 5 of this document includes findings, recommendations, trades, and risks associated with this design of standard attributes for representing relationships.

4.3.1 The Model of Attributes for Representing Relationships

Each *Freedom* object is subject to attributes for representing relationships. The objects and their attributes are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP 30261]). This data standard refers to an applicable document, the SMI (ISO, 1991 [10165]). SMI part 2 contains the ISO attributes used for relationships as described in ISO/IEC 10164-3, *Information Processing Systems - Open System Interconnection - System Management - Part 3: Attributes for*

²⁵ Relationship: An OSI relationship is a set of rules that describe how the operation of one part of a system, element, or payload affects the operation of other parts. A relationship is said to exist among managed objects when the operations of one managed object affects the operation of the other managed objects. For a relationship to be significant within the context of managing the operation of *Freedom*, sufficient information must be available to allow Tier 1 to identify the managed objects involved and the rules governing their interaction.

representing Relationships. The ISO/IEC 10164-3 standard provides more detail on the attributes for representing relationships, and it consistently defines terms that comply with the basic reference model (ISO, 1984 [7498]), the Open System Management Framework (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), the Open System Management Overview (ISO, 1991 [10040]), and the other parts of ISO/IEC 10164.

The standard ISO/IEC 10164-3 defines and explains standards for direct, indirect, symmetric, and asymmetric relationships between two objects.

The standard also identifies three categories of relationships: containment, reciprocal, and one-way. A containment relationship is defined in ISO/IEC 10165-1 and is used for naming and transitioning the managed object to a dormant state. The Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP30261]) uses the containment relationship for the naming of *Freedom's* objects. ISO/IEC 10164-3 defines reciprocal relationships as a special reciprocal binding between two managed objects that point to each other. ISO/IEC 10164 defines the one-way relationship as a special binding that points in one direction.

The standard defines five types of reciprocal relationships: service, peer, fallback, back-up, and group. All of these five types could be used to describe the relationships among *Freedom's* objects, systems, elements, and payloads. The service relationship is an asymmetric relationship denoting rules of service between the first of a pair of managed objects that is providing service to the second object that is using the service. A peer relationship is a symmetric relationship describing the rules of communication between pairs of similar managed objects. The fallback relationship is an asymmetric relationship denoting the second of a pair of managed objects (the secondary object) capable of serving as a fallback or "the next preferred choice" to the first managed object. The back-up relationship is an asymmetric relationship denoting the back-up rules between a pair of managed objects. The second of a pair of managed objects (the back-up object) is providing back-up to the first object (the backed-up object). The back-up object is in the object **disabled** state. The group relationship provides the rules related to the membership of an object to a group of objects. The group relationships are used to relate many objects from the same or different classes to some identified functional or administrative use.

In order to manage the relationships, managed objects are instantiated with the attributes that have values representing the rules of the relationship between the managed objects. The SMI standard (ISO/IEC 10165-2) defines the syntax of these attributes for relationship management. The ISO/IEC 10164-3 standard describes the semantics of these relationship attributes. The ISO/IEC 10165-2 standard includes the following relationship attributes:

The **provider object** attribute can be included in all objects that provide services, and it identifies one or more managed objects acting in a service-provider role with respect to the managed object receiving provider services. The attribute also identifies the order in which service is provided. If the same priority is applied to more than one managed object, then the order of priority is a local matter. The provider object attribute is set-valued and read-write.

The **user object** attribute can be included in all objects that use services of other objects. The user object attribute identifies one or more managed objects acting in a user role with respect to the managed object. The attribute also identifies the order of user priority. If the same priority is applied to more than one managed object, then the order of priority is a local matter. The user object attribute is set-valued and read-write.

The **peer** attribute is used in a managed object definition to identify another managed object that acts in the peer role with respect to the managed object. The peer attribute is single-valued and read-only.

The **primary** attribute is used in a managed object definition to identify one or more managed objects acting in a primary role with respect to the managed object. The attribute also identifies the order of priority in which they act in a primary role. If the same priority is applied to more than one managed object, then the order of priority is a local matter. The primary attribute is set-valued and read-write.

The **secondary** attribute is used in a managed object definition to identify one or more managed objects acting in a secondary role with respect to the managed object. The attribute also identifies the order of priority in which they act in a secondary role for the defined managed object. If the same priority is applied to more than one managed object, then the order of priority among these managed objects is a local matter. The secondary attribute is set-valued and read-write.

The **backup object** attribute can be included in defined managed objects to identify a managed object acting in a back-up role with respect to it. The back-up object attribute is single-valued and read-only, although its value is null if the managed object that owns the attribute is currently active and not in need of back-up service. The back-up object attribute forms the back-up object parameter defined in the alarm reporting function standard (ISO/IEC 10164-4).

The **backed up object** attribute can be included in defined managed objects to identify a managed object acting in a backed up role with respect to it. The backed up object attribute

is single-valued and read-only, although its value is null if the managed object that owns the attribute is not currently active as a back-up on behalf of any other object.

The **member object** attribute can be included in defined managed objects to identify one or more managed objects acting in the member role with respect to that managed object. The member object attribute is set-valued and read-write.

The **owner object** attribute can be included in defined managed objects to identify one or more managed objects acting in the owner role with respect to that managed object. The owner object attribute is set-valued and read-write.

The **relationships attribute group** comprises all of the relationship attributes of a managed object. The identifier of this group is the same for all objects of all classes. The attribute is set-valued and read-only. If a read service like RODB READ is made of the identifier of this group attribute, then the set of attribute identifiers and the values of the attributes in the attribute group are returned.

4.3.2 Notification of Changed Attributes that Represent Relationships

The relationship change notification is used to report the changes in the value of one or more relationship attributes of a managed object. The changes are either the result of an internal behaviour of the managed objects or as the result of management commands. The relationship Change notification has the parameters shown in Table 3.

4.3.3 Attributes and Objects for Representing Relationships Service Definitions

The attributes and objects for representing relationships will be provided by the detail design of the ISE. Examples of how ISE and DMS could map these attributes and objects into communications' services to be carried by CMIS are provided in the ISO/IEC IS 10164-3. The designs of the ISE and DMS do not have to comply with ISO/IEC IS 10164-3 but the functions of the space station relationship management will require equivalent services and relationship information.

4.3.4 Attributes for Representing Relationship's Protocol and Abstract Syntax Definitions

The attributes for representing relationships are defined and explained by the ISO/IEC IS 10164-3. The ISO/IEC IS 10165-2, defines the abstract syntax for the relationship attributes.

Table 3. Parameter of the Changed Attributes of a Relationship Change Notification

PARAMETER	Mandatory/Optional /Conditional
source Indicator	Mandatory
notification type	Mandatory
event time	Optional
list of parameters corresponding to the relationship attributes whose values change are being reported and the additional info parameter	Mandatory
old value of the user object attribute	Optional
current values of the user object attribute	Conditional
old value of the provider object attribute	Optional
current values of the provider object attribute	Conditional
old value of the peer attribute	Optional
current values of the peer attribute	Conditional
old value of the primary attribute	Optional
current values of the primary attribute	Conditional
old value of the secondary attribute	Optional
current values of the secondary attribute	Conditional
old value of the backup object attribute	Optional
current values of the backup object attribute	Conditional
old value of the backed-up object attribute	Optional
current values of the backed-up object attribute	Conditional
old value of the owner attribute	Optional
current values of the owner attribute	Conditional
old value of the member attribute	Optional
current values of the member attribute	Conditional
correlated notifications	Optional
additional text	Optional
additional state change info	Optional
current time	Optional

The ISO/IEC IS 10165-3 specifies the following attributes for representing relationships:

- additionalInfo
- providerObject
- userObject
- peer
- primary
- secondary
- backUpObject
- backedUpObject
- member
- owner

The ISO/IEC IS 10165-2 specifies the relationshipChangeRecord as a supporting object for representing relationships. The relationship change record is needed to support the logging of the relationship changes. Section 4.6 of this document describes the log control function.

4.4 Alarm Reporting Function

This section of the document describes the alarm²⁶ report function that may be used by the application processes in the Space Station *Freedom* Program. This alarm reporting function is proposed to meet the requirements of the SSFP Tier 1 to monitor and control C&W alarms such as fire, smoke, pressure levels, and other threshold levels set by either Tier 1 management or a managed object (system, element, or payload). The ISE as part of Tier 1 needs to understand alarm reports²⁷ (notifications) and needs to have established agreements in the form of interface control documents (ICDs) that specify the standards used to provide the types of alarm notifications and standard meanings of the reported parameters²⁸. For example, standard alarm types could be related to communications, quality of service,

²⁶ Specified alarm: An ISO/IEC specified alarm is a notification of the form defined by the ISO/IEC IS 10156-4 alarm function and a specific event. An alarm may or may not represent an error.

²⁷ Alarm report: An alarm report is a specific type of event report used to convey alarm information.

²⁸ Parameter of a notification: A parameter of a notification is the reported bit field that is to be filled with an attribute value. The coding of the attribute value in the parameter is to follow the standardized ASN.1 transfer syntax as specified in the ISO/IEC IS 10165-2.

equipment, environment, safing, mode changes, etc. These alarm types could be prioritized and discriminated for action by Tier 1 management operations. The identification and standardization of probable cause information would aid the analysis of the alarm root cause. Also, the correlation of alarm notifications could relate the alarms and aid the resolution of the problems. The logging of alarms is also important, so the identification and standardization of logging records would aid in managing the alarms.

A standard alarm management function that provides these basic needs would form a part of a systematic and flexible command structure. The following sections include a description of the alarm management function as standardized by ISO/IEC. This alarm management function meets the needs of the Tier 1 components. Section 5 of this document includes findings, recommendations, trades, and risks associated with this design of a standard alarm management function.

4.4.1 The Model of the Alarm Function

Each *Freedom* object is provided a DMS STSV for the generation of C&W alarm notifications. The objects with their attributes, their notifications, and the attributes to be monitored by the DMS C&W STSV are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP 30261]). This data standard refers to an applicable document the SMI, (ISO, 1991 [10165]). SMI part 2 contains the ISO attributes and objects used for alarm reporting as described in ISO/IEC 10164-4, *Information Processing Systems - Open System Interconnection - System Management - Part 4: Alarm Reporting Function*. The ISO/IEC IS 10164-4 standard provides a standard way of reporting alarms, errors and related information. The SMI standard (ISO, 1991 [10165]) defines syntax for the attributes, objects, and the generic notification of the alarm function. The ISO/IEC 10164-4 standard describes attributes, objects, and the generic notification of the alarm function. The ISO/IEC standard 10164-4 provides detail on the attributes and objects for the alarm reporting function, and it consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), the *Open System Management Overview* (ISO, 1984 [7498-1]), and the other parts of ISO/IEC 10164. In addition, the committee draft (CD) standard on workload monitoring (ISO, 1991, [CD 10164-11]) provides a model of the thresholds used to generate alarm notifications.

The standard model in ISO/IEC 10164-4 contains criteria for reporting the severity of the alarms, types of alarms concerning detected faults or abnormal conditions, a way to include correlations of alarms in notifications, a common set of types of notifications, and records for logging alarm notifications. The standard specifies five alarm notification types and their parameters and semantics. The standard also gives a generic set of values and guidance as to

how the information concerning the alarm can be categorized. In addition, the standard defines an alarm record object that allows alarm notifications to be logged as specified in ISO/IEC 10164-6.

The ISO/IEC 10164-4 standard defines a generic alarm notification that reports five generic alarm notification types.

- **Communication alarms** associated with the behaviours required to convey information from one point to another.
- **Quality of service alarms** associated with degradation in the performance of a service.
- **Processing alarms** associated with a software or processing fault.
- **Equipment alarms** associated with an equipment fault.
- **Environment alarms** associated with a condition related to an enclosure in which the equipment or managed objects reside.

The ISO/IEC 10164-4 standard defines a mandatory parameter that qualifies the **probable cause** of the alarm. The standard defines and registers a set of probable causes that have wide applicability. The standard specifies that the probable cause values are to be indicated in the behaviour clause of the object class definition. The standard defines and explains the following probable causes (possible *Freedom* applications are listed first):

- Fire
- Smoke detection
- Enclosure door open
- High/low ambient temperature
- High/low humidity
- Heating/cooling system failure
- Ventilation system failure
- Toxic gas
- High/low pressure
- Power problem
- Pump failure
- Loss of signal
- Framing error
- Bandwidth reduced
- Retransmission rate excessive
- Threshold crossed
- Storage capacity problem
- Version mismatch (Configuration Management Error)
- Corrupt data
- CPU cycles limit exceeded
- Software error
- Out of memory
- Underlying resource unavailable
- Timing problem
- Processor problem
- Local transmission error
- Remote transmission error
- Call establishment error
- Degraded signal
- Response time excessive
- Queue size exceeded
- Terminal problem
- External interface device problem
- Dataset problem
- Multiplexer problem
- Receiver failure
- Transmitter failure
- Trunk card problem

The standard specifies an optional parameter to further refine the probable cause of the alarm notification. The standard calls it the **specific problems** parameter. The standard suggests that further specific problem identifiers should be defined and registered using the registration procedures defined for ASN.1 Object Identifier values in ISO 8824.

The standard specifies six perceived severity levels for conveying the effected capability of the managed object associated with the alarm notification. The standard lists the severity levels as follows:

- The **cleared severity level** indicates the clearing of one or more previously reported alarms. This alarm clears all alarms reported by the managed object of the same alarm type, probable cause, and specific problems parameter. Associated notifications that are correlated may be cleared by using the correlated notification parameter (listed below).

- The **indeterminate severity level** indicates that the effect of the problem can not be determined.
- The **critical severity level** indicates that immediate corrective action is required.
- The **major severity level** indicates that urgent corrective action is required.
- The **minor severity level** indicates that corrective action should be taken to prevent serious failure.
- The **warning severity level** indicates that diagnostic action (if necessary) and corrective action to prevent the progression of the fault to a failure should be taken.

The standard specifies an optional **backed-up object instance** parameter. This parameter has the value of the relationship attribute, `backedUpObject`. The use of this parameter in conjunction with the severity parameter allows assessment of the seriousness of the reported fault and the ability of the system as a whole to continue to provide services.

The standard specifies an optional **severity trend** parameter. If present, it indicates there is one or more outstanding alarms that have not been cleared from a managed object. The severity trend indicator has three levels:

- The **more severe** level indicates the current alarm is of higher severity than any of the outstanding alarms.
- The **no change** severity level indicates the current alarm is of the same severity as the most severe of any of the outstanding alarms.
- The **less severe** level indicates the current alarm is of lower severity than at least one of the outstanding alarms.

The standard specifies a required conditional **threshold information** parameter when the alarm is the result of crossing an ISO threshold²⁹. It consists of four subparameters³⁰:

²⁹ **Threshold:** An ISO threshold is modeled as an attribute with two levels: the triggering level and the clear level. Each of these threshold levels may be triggered on either an increasing or decreasing gauge values. By setting the attributes of the two threshold levels and values of the decreasing and increasing attributes, an ISO threshold can be functionally set to have a specified hysteresis (the difference between the triggering level and the clear level) and trigger or clear only on the first crossing in an increasing or

- The **triggered threshold** attribute value identifies the threshold attribute that caused the alarm notification.
- The **threshold level** attribute value consists of either a gauge threshold value and the gauge hysteresis value or a counter threshold value.
- The **observed value** attribute value that crossed the threshold value.
- The **arm time** attribute value is the time when the threshold experienced an object enabled event. (For resettable counter thresholds, the object enabled event occurs at each reset.)

The standard specifies an optional **notification identifier** parameter that identifies the alarm. The correlated-notification parameter contains this parameter in future notifications. The notification parameters are chosen to be unique across all notifications of a particular managed object (system, element, or payload) throughout the time that correlation is significant.

The standard specifies an optional **correlated-notifications** parameter that contains subparameters to identify the set of notification identifiers and, if necessary, their associated managed object instance names. This set is defined to be the set of all notifications with which the alarm is correlated. The source object instance value is mandatory if the correlated event report is from a managed object instance other than the one in which the correlated-notification parameter appears. The correlation algorithm is accomplished by the agent system and is not specified.

The standard specifies an optional generic state change parameter when there is a state transition as specified in ISO/IEC IS 10164-2. This parameter has two subparameters:

decreasing direction. ISO/IEC IS 10165-2 specifies these threshold attributes and their values. ISO/IEC CD 10164-11, clause 7, provides the generic threshold model description.

- ³⁰ Subparameter: A subparameter of a notification parameter is an included bit subfield that is to be filled with an attribute value. The coding of the attribute value in the subparameter is to follow the standardized ASN.1 transfer syntax as specified in the ISO/IEC IS 10165-2.

- The **generic old state** parameter has the value of the object state at the time the alarm occurred.
- The **generic new state** parameter has the current value of the object state.

The standard specifies an optional **monitored attributes** parameter that identifies one or more attributes of the managed object and their corresponding values at the time of the alarm. The space station managed object identifiers may specify the set of attributes that are of interest, if any. This allows timely reporting of changing conditions prevalent at the time of the alarm.

The standard specifies an optional **proposed repair action** parameter that the managed object (system, element, or payload) can suggest. This parameter is a set of an enumerations of possibilities specified in the object class definition.

The standard specifies an optional **additional text** parameter that is a free form field text description of the alarm and the problem reported. The standard does not specify the format or the meaning of the data content in the problem text parameter.

The standard specifies an optional **additional information** parameter that allows the inclusion of a set of additional information in the alarm notification. The problem data parameter is a series of data structures that contain three items of information: an identifier, a significance indicator, and the problem information. The identifier subparameter has the value of a registered object identifier that defines the data type of the information subparameter. The significance subparameter indicates whether the receiving system must parse the contents of the information subparameter of the alarm report. The information subparameter carries the information about the problem.

All of the standards optional and mandatory alarm notification parameters should be considered in the DMS C&W STSV.

4.4.2 Notifications of the Alarm Function

The alarm notification is used to report the problems of a managed object. The alarms are either the result of internal behaviour of the managed objects or the result of management commands. The alarm notification of the alarm function has the parameters discussed in the modeling section of the alarm notifications. The standard specifies the five alarm function notification types as separate alarm reporting notifications. The five specified alarm reporting notifications are as follows:

- communicationAlarm
- qualityOfServiceAlarm
- processingAlarm
- equipmentAlarm
- environmentalAlarm

The ISO/IEC IS 10164-4, clause 11, maps the parameters of the notifications to the CMIS parameters.

4.4.3 Attributes and Objects for Alarm Function Service Definitions

The attributes and objects for representing relationships will be provided by the detail design of the DMS, ISE, and the *Freedom's* objects (systems, elements, and payloads). The DMS C&W STSV should have notifications to meet the needs of the SSFP. DMS STSV should provide the notification parameters and managed objects of the standardized alarm notification function. Examples of how DMS could map these notification parameters, attributes, and objects into communications services to be carried by CMIS are provided in the ISO/IEC IS 10164-4. The designs of the ISE and DMS do not have to comply with ISO/IEC IS 10164-4 but the capability of DMS STSV will require the functions of the standard.

4.4.4 Attributes for the Alarm Function and Abstract Syntax Definitions

The attributes for the alarm function are defined and explained by the ISO/IEC IS 10164-4. The ISO/IEC IS 10165-2, defines the abstract syntax for the alarm function attributes.

The ISO/IEC IS 10165-3 specifies the following attributes for representing relationships:

- probableCause
- specifiedProblems
- perceivedSeverity
- backUpStatus
- backUpObjectInstance
- tendIndication
- thresholdInfo
- notificationID
- correlatedNotifications
- genericStateChange
- monitoredAttributes

- proposedRepairActions
- problemText
- problemData

The ISO/IEC IS 10165-2 specifies the eventLogRecord and alarm record as supporting object for the alarm function. The event log record is needed to support the logging of the alarms. Section 4.7 of this document includes a description of the log control function.

4.5 Event Reporting Function

This section of the document describes the event reporting function that may be used by application processes in the Space Station *Freedom* Program. The event reporting function is proposed to meet the requirements of the SSFP Tier 1 to monitor and control the transmission of event notifications from the managed object independent of the definition of the managed object (system, element, or payload). The ISE as part of Tier 1 needs to have a flexible event report control service that allows systems to select which event notifications are to be sent to particular managing systems. (For example, FDIR events to ISE, to a ground FDIR workstation, and/or to a ground system controller's workstation.) Tier 1 needs controls to specify the destinations to which the event notifications are to be sent. Tier 1 needs controls to suspend and resume the forwarding of or sending of the event notifications. Tier 1 needs controls to modify the conditions used in reporting events. Tier 1 needs controls to designate a back-up location to which event notifications can be sent if the primary location is not available.

In addition, the processing of event notifications needs a flexible discriminator that filters event notifications for specific actions. The Tier 1 components (and in particular ISE) need controls to set and modify the selection of alarm notifications, especially the C&W alarm notifications reported from DMS STSVs. The Tier 1 components need a discriminator that compares the values of the C&W alarm notification parameters, and then, based on the comparison, selects and commands a sequence of predetermined alarm actions. The Tier 1 components (and in particular ISE) need controls to set and modify the selection of FDIR notifications that report significant failures. The Tier 1 components need a discriminator that compares the values of the FDIR event notification parameter, and then, based on the comparison, selects and commands a sequence of predetermined failure isolation and failure recovery actions. The Tier 1 components (and in particular ISE) need controls to set and modify the selection of station mode change notifications, object creation notifications, object deletion notifications, object name change notifications, attribute value change notifications, state change notifications, and relationship change notifications. The Tier 1 components need a discriminator that compares the values of these notifications, and then, based on the comparison, selects and commands a sequence of predetermined ISE actions. The ISE actions based on the information in the notifications would prepare for mode

transitions, perform mode transitions, and maintain the station mode status. The ISE actions would check station resource constraints and command constraints.

A standard event management function that provides these basic needs would form a part of a systematic and flexible command structure. The following sections include descriptions of the event management function standardized by ISO/IEC. This event management function meets the needs of the Tier 1 components. Section 5 of this document includes findings, recommendations, trades, and risks associated with this design of a standard event management function.

4.5.1 The Model of the Event Reporting Function

Each *Freedom* object needs a DMS STSV for the control of event notifications. The objects, systems, elements, and payloads, with their attributes and their event notifications, will send messages to Tier 1 components. The objects, systems, elements, and payloads are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP 30261]). This data standard refers to an applicable document the SMI (ISO, 1991 [10165]). SMI part 2 contains the ISO attributes and objects used for managing event notifications as described in ISO/IEC 10164-5, *Information Processing Systems - Open System Interconnection - System Management - Part 5: Event Reporting Function*. The ISO/IEC IS 10164-5 standard provides a standard way of managing event reports. The SMI standard (ISO, 1991 [10165]) defines syntax for the attributes, objects, and the generic notification of the event reporting function. The ISO/IEC 10164-5 standard describes how the attributes and objects of the event management function work together to control event notifications. The ISO/IEC standard 10164-5 provides detail on the attributes and objects for the event reporting function, and it consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the CMIS (ISO, 1989 [9595]), the *Open System Management Overview* (ISO, 1991 [10040]), and the other parts of ISO/IEC 10164.

The standard ISO/IEC 10164-5 specifies the event reporting function that serves as a generic discriminator that filters event notification for use by management. The ISO/IEC 10164-5 standard defines a generic event notification management model. Figure 7 illustrates the OSI event forwarding discriminator model. This model includes components for the reporting of events to the SSCC and POIC and the on-board processing of potential event notifications. The model also describes the message flow of event reporting function control messages from the SSCC to the event reporting function discriminator. The model indicates that all event notification onboard from the objects, systems, elements, and payloads are reported to a service detecting and processing the event notifications. Conceptually these potential event notifications are distributed to all event forwarding discriminators that are contained within on-board DMS. The event forwarding discriminator determines which event notifications

are to be reported to a particular destination during specified time periods. Each event forwarding discriminator contains scheduling capability (or uses a scheduling function, see section 4.10) to determine the interval during which event notifications will be selected for forwarding. Each event forwarding discriminator contains a discriminator construct that specifies the characteristics that a potential event notification must satisfy in order to be forwarded or processed. Event notifications that pass the discriminator are sent to the destination as soon as possible.

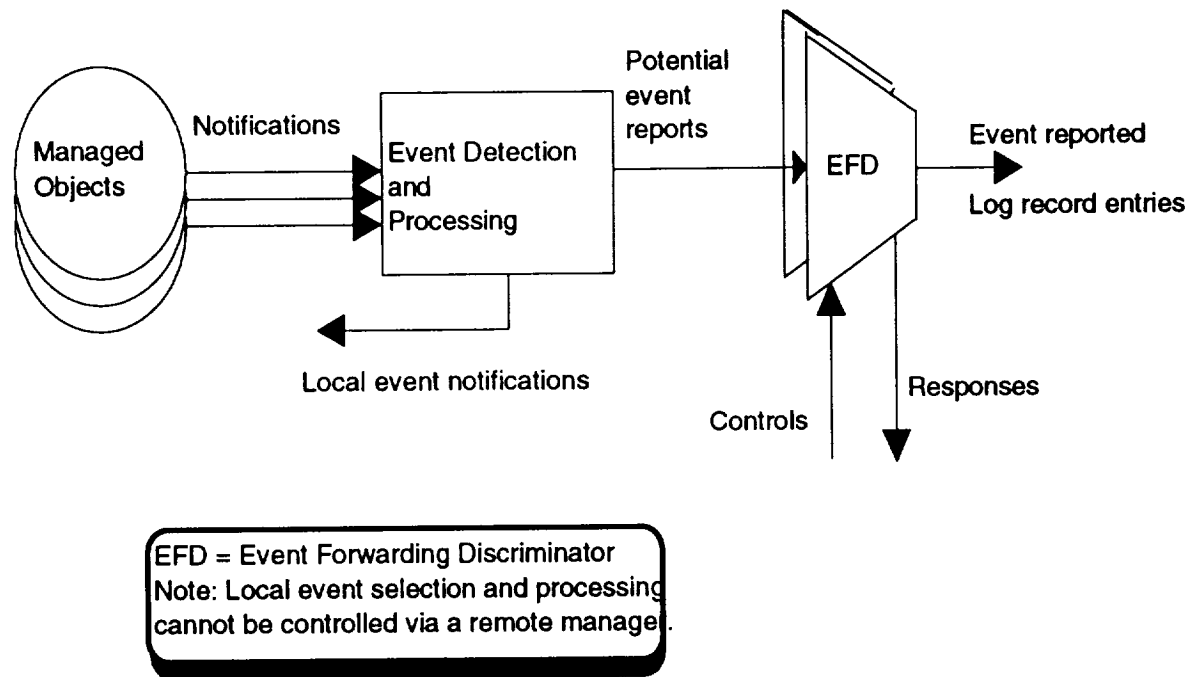


Figure 7. The Event Report Management Model

The ISO/IEC 10164-5 standard specifies the following event reporting function services.

- Initiation of event forwarding
- Termination of event forwarding
- Suspension of event forwarding
- Resumption of event forwarding
- Modification of event forwarding conditions
- Retrieval of event forwarding conditions

The standard specifies that event notifications may be reported to the managing system, (like the SSCC or ISE) or a third system (like the POIC). The event reporting managing function

provides the capability for establishing a long-term event reporting relationship between the managed system and managing systems. While the event forwarding discriminator is in the administrative unlocked state, the reporting system forwards the event notifications to the specified notification.

The specified event reporting function is so general that it applies equally to the transfer of a logging notification and the transfer of summarization notifications (TOLs). The ISO/IEC CD standard 10164-13 specifies summary notifications (TOLs) that are also controlled by the event reporting function. The ISO/IEC standard 10164-6 specifies the logging function. The logging function notifications are also controlled by the event reporting function. Thus the services of initiation, termination, suspension, resumption, modification of conditions, and the retrieval of conditions apply to logging and summarization notifications.

The standard specifies a basic superclass object to define the discriminator behaviour. The discriminator may be defined into subclasses to specify management support object classes that allow the control of various system management functions. The discriminator provides for the specification of conditions that must be satisfied prior to allowing the associated discriminator input to be processed. Some of the conditions are common to all subclasses of the management service control discriminator, others are unique to the specific discriminator subclass.

The conditions specified by the management service control discriminator are:

- the identification of a scheduling package³¹ that determines when event forwarding will occur
- the criteria for discrimination
- the administrative, usage and operational state of the discriminator
- those conditions specified as specific to a particular discriminator object subclass.

The ISO/IEC 10164-5 standard describes the management of discriminator objects as the generic process that exercises control over the management operations and notification of any other managed system. Discriminators can be created, deleted, read, and modified by communicating with the discriminator objects. In addition, the activity of the discriminators can be suspended and resumed by means of manipulating their administrative state.

³¹ Package: An OSI package is a set of optional behaviours with associated attributes that are imported into the managed object at the time the managed object was created or initialized. A conditional package is one of a set of packages that is selected based upon conditional attributes supplied at the time the managed object was created or initialized.

Thus, the standard specifies that discriminator objects can control practically any other object in a managed system. If, for example, the discriminator is used as the means of controlling the command constraint list, then the override commanding of the access control object implementing the command constraint is accomplished with a discriminator suspend or terminate service.

The ISO/IEC 10164-5 standard specifies that each discriminator has administrative, operational, and usage states. The administrative, operational, and usage state, and availability status defined for the discriminator are subsets of the corresponding attributes defined in ISO/IEC 10164-2 (see section 4.2).

The operation states specified for the discriminator are enabled and disabled. The discriminator process inputs when enabled and the availability status is not "off-duty." The discriminator process does not process input when disabled.

The standard specifies the locked and unlocked attribute values of the administrative state for the discriminator. The operation of the discriminator in the administrative state depends upon standardized subclasses of discriminators. The managing system (a Tier 1 component) may lock or unlock the discriminator. When the administrative state of the discriminator changes, the discriminator generates a notification. When the state is changed from unlocked to locked, the discriminator first processes the notification and then makes the change. When the discriminator is changed from locked to unlocked, the discriminator first makes the state change and then processes the notification without disrupting the reporting of a current potential event notification.

The standard-specified discriminator generates a deletion notification indicating deletion prior to executing a deletion command.

The standard specifies that the time when the discriminator is available (i.e., "on-duty") can be changed. The standard also specifies that the conditions the discriminator evaluates can be changed. These changes are defined to occur in a way so as to not affect the processing of the discriminator inputs being processed.

The standard specifies the operation of discriminators by the use of a discriminator construct that is a filtering mechanism. The discriminator construct acts on the attributes of the discriminator inputs. A discriminator construct is a set of one or more assertions about the presence or values of attributes. If the discriminator construct involves more than one assertion, then the assertions are grouped together using a logical operator.

The standard specifies the following test:

- Equality
- Inequality
- Presence of attributes
- Negation of equality, inequality, or presence of attributes
- Multiple constructs using the operators:
 - AND
 - OR

The standard specifies the behaviour of the discriminator object. If an asserted attribute is absent, then the discriminator evaluates as **FALSE**. If the discriminator construct is empty and all other conditions (such as time) are satisfied, then the discriminator evaluates as **TRUE**. If the discriminator construct evaluates to **TRUE**, the discriminator is in the object unlocked and object enabled states, and the availability status is not "off-duty", then the discriminator input passes the discriminator and will be processed further (the additional behaviour depends upon the behaviour specified by the definition of the discriminator subclass). If the discriminator is in the locked state or has the off-duty availability status, then the discriminator inputs will not be processed by that discriminator.

The standard specifies the following mandatory attributes for the discriminator superclass.

- The **discriminator ID** identifies the instance of the discriminator object.
- The **discriminator construct** specifies tests on the information to be processed by the discriminator.
- The **administrative state** attribute indicates the object unlocked and object locked states.
- The **operational state** attribute indicates the object enabled and object disabled states.
- The **availability state** attribute indicates the off-duty condition.
- The **mode** attribute indicates the specification of any required mode for reporting events. (For example, some event reports may not be confirmed for a station mode.)

The standard specifies discriminator scheduling packages, one of which may be included within the discriminator when it is created. The scheduling packages are used to

automatically switch between their reporting periods. To accommodate various levels of complexity in scheduling of event reporting activity periods, three conditional schedulings are defined for the event forwarding discriminator. One conditional package provides for schedule reporting within a 24 hour period, the second provides for schedule reporting within a period of a week, and the third provides capability for pointing and using a scheduler that is external to the discriminator object. The standard specifies the attributes of the daily and weekly scheduling packages. The attributes of the packages are **IntvlsOfDay**, **StartTime**, **StopTime**, **WeekMask** (consisting of **DayOfWeek** and **IntvlsOfDay**), and the **SchedulerName**.

The standard specifies the mandatory attributes of the event forwarding portion of the discriminator:

- The **destination address** attribute specifies a primary single or a group address.
- The **backup address** attribute specifies an order list of secondary single or group addresses.
- The **active address** attribute specifies the current active address to which the inputs to the discriminator are being reported.
- The **allomorphic list** attribute³² specifies an order list of object classes that will be associated with an event report to be transmitted.

4.5.2 Notifications of the Event Reporting Function

The event reporting function uses event notifications to report changes in the event reporting function. The standard specifies four event reporting function notifications. The four used event reporting notifications are as specified by ISO/IEC 10164-1, Object Management Function:

- State change notification
- Attribute value change notification
- Object creation notification

³² Allomorphism: Allomorphism is the ability of a managed object that is an instance of a given class of managed object to be managed by one or more other managed object classes. An ordered allomorphic list is a sequence of associated objects related by modifications in behaviour. For example, a revised managed object that operates as the unrevised managed object is allomorphic.

- Object deletion notification

The ISO/IEC IS 10164-1, clause 11, provides the mapping of the parameters of the notifications to the CMIS parameters.

4.5.3 Attributes and Objects for Event Reporting Function Service Definitions

The attributes and objects for representing the event reporting function will be provided by the detail design of the DMS, ISE and the *Freedom's* objects, systems, elements, and payloads. The DMS STSV should have an event reporting function to meet the needs of the SSFP. DMS STSV should provide the objects and attributes of the standardized event reporting function. Examples of how DMS could provide the event reporting functions are provided in the ISO/IEC IS 10164-5. The designs of the ISE and DMS do not have to comply with ISO/IEC IS 10164-5 but the capability of DMS will require the functions of the standard.

The event forwarding discriminator provides the services of creating a discriminator, deletion of a discriminator, modifications of discriminator attributes, suspension of the activity of the discriminator, and resumption of the discriminator activity.

The attributes for the event reporting function defined and explained by the ISO/IEC IS 10164-5. The ISO/IEC IS 10165-2, defines the abstract syntax for the event reporting function attributes.

The ISO/IEC IS 10165-5 specifies the following attributes for the event reporting function:

- discriminatorID
- discriminatorConstruct
- allomorphicList
- availabilityStatus
- destinationAddress
- backupAddress
- administrativeState
- operationalState
- StopTime
- StartTime
- objectClass
- mode (event modes)
- DaysOfWeek

The ISO/IEC IS 10165-5 specifies the **eventReportRecord** and **eventForwardingDiscriminator** as supporting objects for the event reporting function. The event report record is needed to support the logging of the events reports processed and sent by the event forwarding discriminator object. Section 4.6 of this document describes the log control function.

4.5.4 Event Function Protocol and Abstract Syntax Definitions

The ISO/IEC 10165-2 specification defines the ASN.1 value notations for all the objects and attributes needed by the event reporting function.

4.6 Objects and Attributes for Access Control Management

This section of the document describes the objects and attributes for access control³³ that may be used by application processes in the Space Station *Freedom* Program. The objects and attributes for access control management are proposed to meet the requirements of the SSFP Tier 1 to monitor and control command constraints and other management operations such as controlling who should receive FDIR alarm reports and controlling security access to payload data. The ISE as part of Tier 1 needs flexible access control service that allows Tier 1 to monitor and control access to command constraints, notification reporting address, log records, and the media data³⁴ of access control. The SSFP Tier 1 also needs the ability to have a consistent set of definitions and actions related to management of the access control commands that *Freedom's* systems, elements, and payloads receive. In cooperation with the *Freedom* object management function (see section 4.1) and state management function (see section 4.2) and the service control functions, Tier 1 needs the ability to manage the access controls.

Each *Freedom* object needs a DMS STSV for access control. The objects, systems, elements, and payloads with their attributes and their event notifications require that unauthorized access to the applications and management information be prevented by the use

³³ Access control: Access control is defined by ISO-7498-2 as the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.

³⁴ Media data: In this document, media data is data about data. Media access control data is the attribute values of attributes associated with access control object, e.g., security policy rules, authorized list of controllers and commanders, list of actions related to transitioning between station modes, etc.

of one or more access control mechanisms. Control of access to station information is required (or needed) in each of the following cases:

- To prevent unauthorized establishment of associations
- To protect station information from unauthorized creation, deletion, modification or disclosure by means of a commanded operation
- To prevent unauthorized initiators from using command and control operations; (i.e., to ensure that only the necessary privileges are obtained during the application association)
- To prevent management information from being transmitted to unauthorized recipients by means of confirmed and non-confirmed event notifications.

The various levels of access control may be required. For example, different command constraints would be applicable to different station modes. For command and control, access restrictions must cater to the managed objects, systems, elements, and payloads; individual attributes of managed objects, systems, elements, and payloads; individual actions of objects, systems, elements, and payloads; and individual values of individual attributes.

Access control of Tier 1 commands require the ability to cater to any and all of the security policies³⁵, security service, and mechanism applicable to access control. And in order to facilitate and access the control information for Tier 1, it is desirable that access control information be modeled as managed objects so that all the other object oriented services can be applied to the access control objects.

Thus, Tier 1 needs a flexible access control service that allows selection of access control information³⁶. Tier 1 needs the ability to modify the criteria used in allowing access control to any protected entity³⁷. Tier 1 needs the ability to determine whether the access controls

³⁵ Security policy: Security policy is defined in ISO 7498-2 as the set of criteria for the provisions of security services.

³⁶ Access control information: Access control information (ACI) is any information that is used for access control purposes.

³⁷ Protected entity: A protected entity is a set of one or more items that are protected by the same access restrictions. An item may be a software application process, a set of

were modified. Tier 1 needs a mechanism to control the time during which access occurs, (for example, suspending specific access during selected station modes). Tier 1 needs the ability to retrieve and delete selected access control policies³⁸. And Tier 1 needs the ability to initialize and enable access control object (create) and shutdown, terminate, and make dormant (delete) access control objects.

Standard access control objects and attributes that provide these basic needs would form a part of a systematic and flexible command structure. The following sections include descriptions of the objects and attributes for access control proposed for standardization by the committee draft, ISO/IEC 10164-9. These objects and attributes for access control meets the needs of the Tier 1 components. Section 5 of this document includes trades, issues, and risks associated with supplying a DMS STSV with this design of objects and attributes for access control.

4.6.1 The Model of the Objects and Attributes for Access Control

The objects, systems, elements, and payloads are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP30261]). This data standard refers to an applicable document, the SMI, ISO/IEC 10165. SMI part 2 will contain the ISO attributes and objects used for managing access control when ISO/IEC 10164-9, *Information Processing Systems - Open System Interconnection - System Management - Part 9: Objects and Attributes for Access Control*, becomes an international standard. The model of the access control objects has become very stable, only the number and types of access control attributes are changing. The basic attributes are unlikely to change, and besides only basic attributes are needed for *Freedom* access controls. Currently the committee draft, ISO/IEC CD 10164-9, defines syntax for the attributes,

attributes controlled by an application process, or a set of values of a single attribute controlled by an application process.

- ³⁸ Access control policy: An access control policy is an aspect of the security policy that is specific to access control. For Tier 1, an access control policy specifies the condition in which security services and mechanisms are used to control access to station information. A SSFP access control policy is a coherent set of rules imposed within the Security domain of the station by the station security authority. A managed object may be in multiple security domains if some aspects of the managed object are under the jurisdiction of different security policies. When a managed object exists in multiple security domains, the enforced access control policy is the one that corresponds to the policy in which the access request originated. By definition, therefore, the initiator and target in any management exchange are governed by the same access control policy.

objects and the generic notification of the objects and attributes for access control. The ISO/IEC 10164-9 committee draft describes how the discriminator construct of the event management function work together to provide the objects and attributes for access control. The ISO/IEC committee draft 10164-9 also consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), the *Open System Management Overview* (ISO, 1991 [10040]), and the other parts of ISO/IEC 10164.

The committee draft ISO/IEC 10164-9 specifies the objects and attributes for access control services for two generic functions: the first function is an access enforcement function (AEF), and the second is an access control decision function (ADF). Figure 8 illustrates the basic access control model. This generic model is based upon the access control model defined in ISO/IEC 10181-3. Authentication of the Tier 1 components is not a part of access control and is outside the scope of 10164-9, but the access control procedures defined in the committee draft assume the use of authentication procedures at the appropriate time.

The committee draft specifies that the information used for access control decision is called access control information (ACI). The committee draft specifies five descriptive terms to explain the uses of the ACI. The draft also specifies three access control policy conditions and three classifications of access control rules. Figure 8 illustrates the five uses of ACI. The five descriptive terms are:

- The **contextual information** is ACI associated with context (e.g., time of day, resource constraints, station mode constraints)
- The **InitiatorADD** is ACI provided by or associated with the initiator³⁹ (the crew [e.g., commander], SSCC [e.g., a workstation function or ground controller] or POIC [e.g., a payload controller or principle investigator] of a request. This information may be conveyed in the access control parameter of the CMIS user information during association establishment, in the access control parameter of the CMIS user information during management operations, or may be preassigned by

³⁹ Initiator: An initiator is the entity that makes the management request for action or information.

the security domain authority. One form of the initiatorADD is the access control certificate⁴⁰ (ACC).

- The **DataADD** is ACI associated with the attributes of managed objects.
- The **RetainedACI** is ACI associated with an initiator. It is retained by the ADF. RetainedACI is used by the ADF to evaluate access privileges whenever a management operation does not contain an ACC.
- The **PolicyACI** is ACI representing the access control policy and relationships in effect within the security domain. PolicyACI is subdivided into two categories:
 - the ACI identifying the protection access control rules
 - the ACI associated with particular targets⁴¹. (This information is represented by managed objects. Target objects may contain permitted or inhibited operations lists that are sets of commands.)

⁴⁰ Access control certificate (ACC): An access control certificate (ACC) is ACI used to specify the access control parameter used with CMIS. The specification of an access control policy may include the definitions of this ACI. For example, ACC may contain:

- the identity of the security domain (e.g, the station, a bank) and the security domain authority (e.g., the SSCC, a bank office)
- the ACI required by access control policy. This information may be one or more of the initiator capabilities (e.g., a system controller, a checking account customer who can deposit and withdraw cash), initiator name (e.g., current commander's name, the personal identification number) or security labels (e.g., crew member certification, ordinary or gold card user).
- the time the ACI becomes valid
- the time the ACI was created
- the integrity check information (e.g., what command constraints are or are not allowed, how big a transaction is allowed)

⁴¹ Target: A target is the entity to which the access request is addressed. (For example, targets are the objects, systems, elements, and payloads that are commanded).

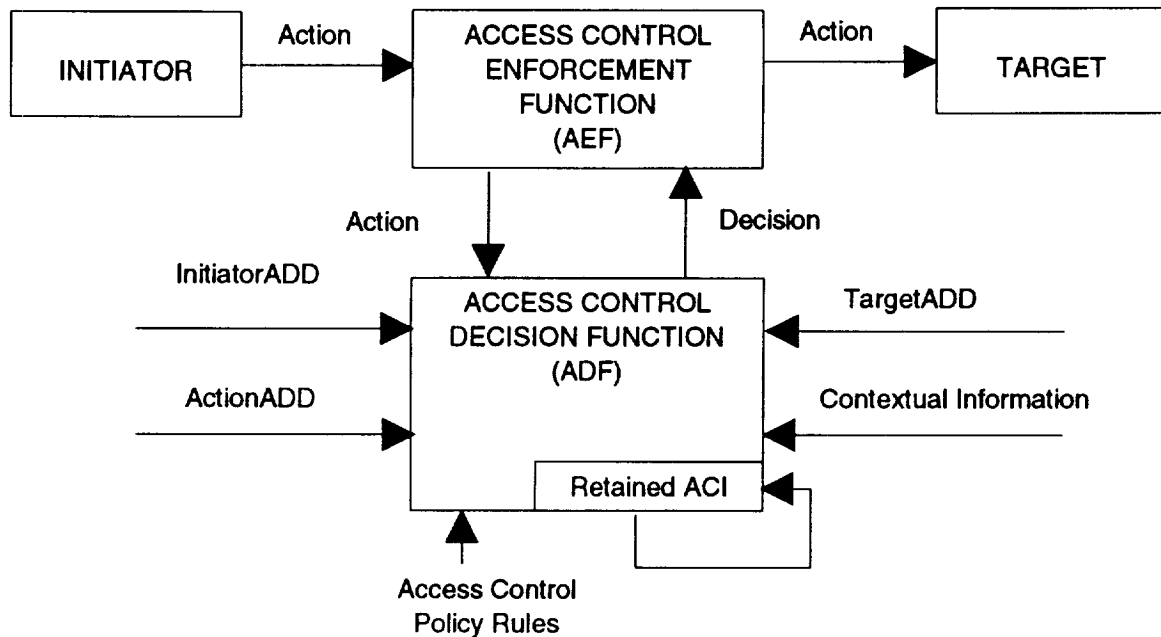


Figure 8. The Generic Access Control Model

The three access control policies represented by the PolicyACI are:

- Access control lists⁴²
- Capabilities⁴³
- Security labeling⁴⁴

⁴² Access control list: Access control lists are defined by ISO/IEC 10181-3 as a mechanism that is used by an identity-based access control policy to specify lists of initiators who may or may not have access.

⁴³ Capability: Access control capability in terms of the access control framework is defined by ISO/IEC 10181-3 as a mechanism used by a capability-based access control policy to specify lists of capabilities that may or may not have access.

⁴⁴ Security label: A security label in terms of the access control framework is defined by ISO/IEC 10181-3 as a mechanism used by a label-based access control policy to specify lists of security clearances that may or may not have access.

The three classifications of access control rules represented by the Policy ACI are:

- Global rules grant or deny access in preference to all other access control rules
- Item rules grant or deny access if no default rule applies
- Default rules grant or deny access if no other rule applies

The model to control access of a management association has the following behaviour in the AEF and ADF (see figure 9).

- First the AEF detects the access control parameter of the CMIS and sends the contained InitiatorADD to the ADF. Then AEF receives the access decision from the ADF and sends the association response related to the access decision and an association response policy.

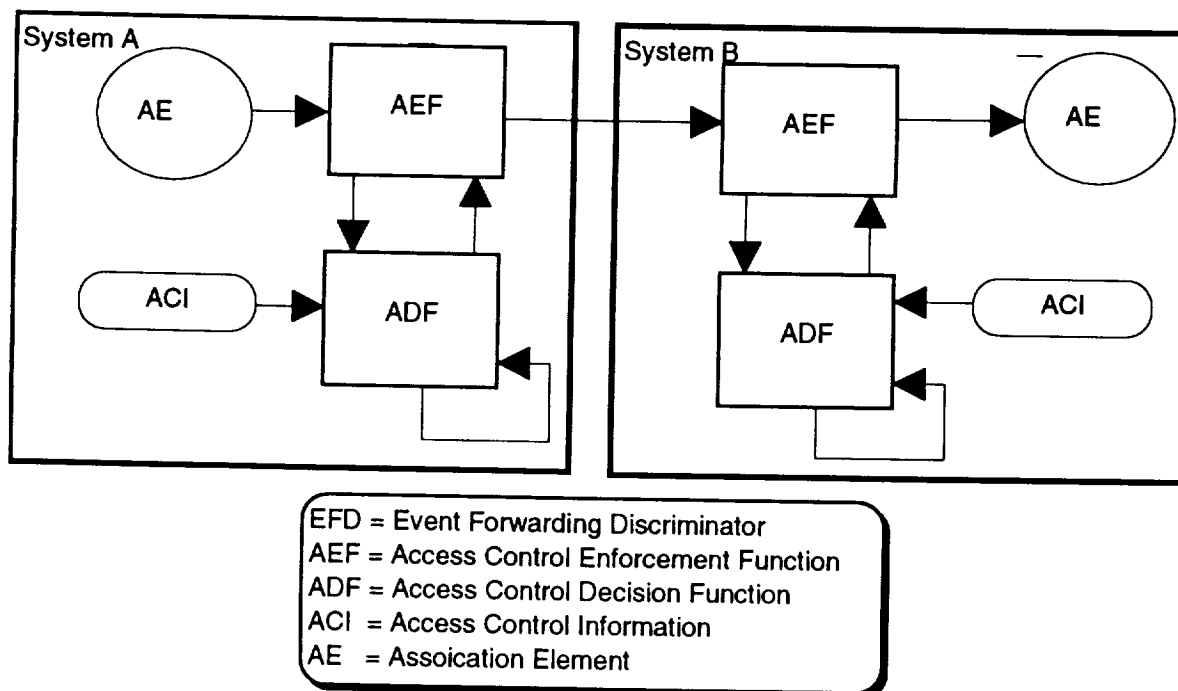


Figure 9. Access Control During Association Establishment

- The ADF evaluates the initiatorADD according to the PolicyACI, to establish if an association is allowed. The ADF sends the association access decision to the AEF.

The model to control access of a management operations has the following behaviour in the AEF and ADF (see figure 10).

- First the AEF detects the access control parameter of each of the CMIS management request (Tier 1 commands) and sends the contained InitiatorADD to the ADF. Then AEF receives the access decision from the ADF and either sends access denial responses to Tier 1, or passes the commands to the managed object (system, element, or payload).

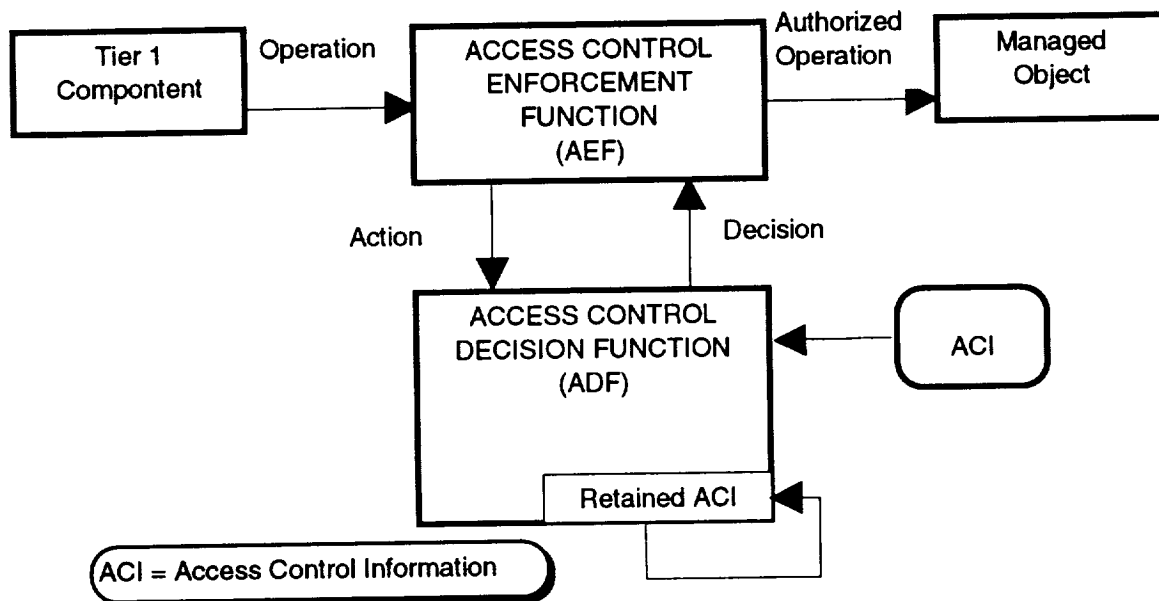


Figure 10. Access Control During Management Operations

- The ADF evaluates the initiatorADD according to the PolicyACI, Target ACI, RetainedACI, DataADD, and contextual information. The ADF sends the management request (Tier 1 commands) access decision to the AEF.

The committee draft specifies a model for notification access policy. This model places the access controls on the reporting of the notification. The ADF uses notification access policy information to control the dissemination of management notifications (see figure 11).

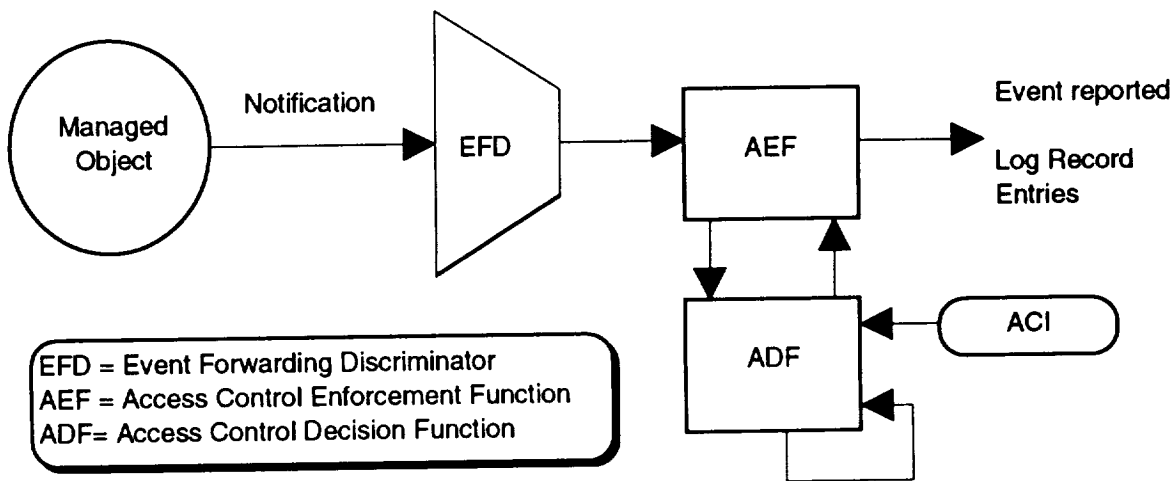


Figure 11 Management Notifications with Applied Access Controls

The committee draft specifies the following three managed objects for access control:

- Access control policy
- Targets
- Authorized initiators

The **access control policy** managed object has the following characteristics:

- It has attributes which define the access control policy rules for association, managed operations, and notifications.
- It has an attribute to identify the security domain.
- It has an attribute to identify the security domain authority.
- It has attributes to provide the validity information, for example, the period in which access is valid.
- It has attributes to provide the object protection information.
- It contains both the **targets** and **authorized initiator** managed objects.

The **targets** managed object has the following characteristics:

- It has attributes to select sets of managed objects by name or filtering.
- It has attributes that identify the operations performed on those sets of managed objects.
- It has attributes that lists the authorized initiators that use the identified operations.

The **authorized initiator** managed object has the following characteristics:

- It has attributes to list the capabilities of initiators.
- It has attributes to detect and match to a security label of the initiators.
- It has attributes to identify features of the initiators.

The committee draft specifies the following access control attributes:

- The **access control list** attribute is used to identify a list of initiators.
- The **access control policy object name** attribute identifies the name of the access control policy managed object.
- The **association access policy** attribute specifies a set of association access control rules.
- The **authentication information** attribute specifies the authentication requirements applied to association initiators.
- The **authorized initiator object name** attribute identifies the name of the initiator managed object.
- The **capability** attribute specifies the capability of initiators.
- The **cryptographic algorithm** attribute defines the algorithm used to encode the access control information in a certificate.
- The **cryptographic checksum** attribute has the value of a checksum of all the information in an access control certificate.

- The **initiator list** attribute lists the authorized initiator managed objects.
- The **notification access policy** attribute specifies the access control policy rules to be applied by the notification access control mechanism.
- The **object access policy** attribute specifies the access policy rules applied to managed objects.
- The **object list** specifies the set of managed objects specified by the **selected object** attribute.
- The **operation** attribute specifies the set operations that may be performed by the authorized initiator on a set of managed objects. (For example, a set of commands that can be enabled or inhibited on a managed object [system, element, or payload].)
- The **security domain attribute** names the security domain and identifies the security domain authority. It may also contain the signature⁴⁵ of the security domain authority to provide assurance that the integrity of this information has not been compromised.
- The **security label** attribute specifies the security label applied to elements of management information.
- The **selected objects** attribute specifies the selection of managed objects to which the access controls are applied.
- The **target list** attribute specifies a list of target managed objects.
- The **targets object name** identifies the target managed object.
- The **time of creation** attribute specifies the time the access control information was created.

⁴⁵ Signature: A signature is defined by ISO 7498-2 as data appended to, or a cryptographic transformation of a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protects against forgery, for example, by the recipient.

- The **valid from** attribute specifies the time at which the access control information becomes valid.
- The **valid until** attribute specifies the time at which the access control information no longer applies.

4.6.2 Notifications of the Objects and Attributes for Access Control

The objects and attributes for access control uses event notifications to report changes in the objects and attributes for access control. The committee draft specifies six access control notifications. The first three used notifications are as specified by ISO/IEC 10164-1, Object Management Function. The last three are specified by the ISO/IEC 10164-7, Security Alarm Reporting Function.

- Attribute value change notification
- Create notification
- Delete notification
- Security service or mechanism violation
- Operational violation
- Time domain violation

The ISO/IEC IS 10164-1, clause 11, and ISO/IEC 10164-7, clause 11, provide the mapping of the parameters of the notifications to the CMIS parameters.

4.6.3 Attributes and Objects for Objects and Attributes for Access Control Service Definitions

The attributes and objects for representing the objects and attributes for access control will be provided by the detail design of the DMS, ISE and the *Freedom's* objects, systems, elements, and payloads. The DMS STSV should have an objects and attributes for access control to meet the needs of the SSFP. DMS STSV should provide the objects and attribute of the standardized objects and attributes for access control. Examples of how DMS could provide the objects and attributes for access controls are provided in the ISO/IEC CD 10164-9 and ISO/IEC IS 10164-7. The designs of the ISE and DMS do not have to comply with ISO/IEC CD 10164-9 or ISO/IEC IS 10164-7 but the capability of DMS will require the functions of the standard.

The attributes for the objects and attributes for access control are defined and explained by the **ISO/IEC CD 10164-9**. The ISO/IEC CD 10165-9 specifies the following attributes for the access control:

- AccessControlList
- AccessControlPolicyObjectName
- AssociationAccessPolicy
- AuthenticationInformation
- AuthorizedInitiatorObjectName
- Capability
- CryptographicAlgorithm
- CryptographicChecksum
- InitiatorList
- NotificationAccessPolicy
- ObjectAccessPolicy
- ObjectList
- Operation
- SecurityDomain
- SecurityLabel
- SelectedObjects
- TargetList
- TargetsObjectName
- TimeOfCreation
- ValidFrom
- ValidUntil

The ISO/IEC 10164-9 committee draft does NOT specify service for manipulation of the access control managed object class.

4.6.4 Protocol and Abstract Syntax Definitions of Objects and Attributes for Access Control

The ISO/IEC 10165-9 committee draft defines the ASN.1 value notations for all the objects and attributes needed by the objects and attributes for access control. These abstract syntax definitions will move to ISO/IEC 10165-2 when ISO/IEC 10164-9 becomes an international standard.

SECTION 5

FINDINGS, RECOMMENDATIONS, TRADE-OFFS, AND RISKS

This document has researched and summarized the current documented requirements for the ISE. A strawman architecture for ISE was then developed to investigate the concept of using the ISO open system management standards in the design of the ISE and to determine the effect on the necessary support functions provided by the DMS. This section of the document presents a summary of the findings and recommendations concerning the ISE requirements. It then provides a discussion of the trade-offs associated with establishing implementor agreements for the adoption of ISO/IEC ISO management standards related to the ten support functions discussed in section 4 of this document. Included in this discussion is MITRE's assessment of the risk in adopting each of the ten management standards for guiding the design of the ISE and DMS.

5.1 Findings and Recommendations Concerning the ISE Requirements

As stated in sections 2, 3 and 4 of this document, the ISE requirements are in need of clarification and expansion into detail specifications. The following paragraphs include MITRE's findings concerning the areas that need improvement.

5.1.1 Findings

Overall assessment -- Current documentation for the ISE in combination with DMS services indicates that NASA will eventually be provided a design that meets most of the intent of the OSI standards. The design shows considerable software modularity, the use of application standard services, and the definition of standard application object classes which may help in overall reduction of the life-cycle costs for the program. There are however, significant differences and departures from the OSI standards that, if left unchanged, will not be compatible with future OSI compliant commercial-off-the-shelf (COTS) software products.

On-board Integration by ISE -- The integration on-board that ISE can perform is minimal (no/limited resource monitoring, no on-board models, predefined command sequences).

- As currently defined by the program, the station and system mode definitions are often overlapping in scope and inconsistent in content. Just as significant, the operational concept for modes is not clearly delineated among the developers, so NASA's intent is not showing up in the system and subsystem designs. The mode concept, which could be a strong aid to integrated operations and safety, is in danger of disappearing as a requirement due to the lack of NASA direction.

- In all documentation reviewed, there was no clear delineation of management requirements among the ISE, DMS, Crew, and SSCC. While it was clear *how* the management would be conducted, *who* had the authority and responsibility for operations management of on-board systems, elements, and payloads was not explicit. For example, it is not clear *who* controls and manages logging, journalizing, scan list, and TOL selection.
- There was not a consistent set of requirements concerning the management of station resources. Some documents require resource management, while recent "verbal reports" indicate resources are not monitored onboard. No models of systems are used onboard, so how resource constraints would affect behaviour of the station is unknown to the on-board systems.
- Not all commands go to the ISE, so the ISE will have a difficult job of integrating the on-board systems. The ISE is limited to executing predefined sequences, for only a limited number of detected on-board configurations.
- There was not a clear definition of the integrating operating requirements as a function of mission build assembly sequence.
- The concept of ISE performing configuration management for the station has been dropped or significantly reduced from the *Level A* to WP-2 documentation. NASA needs to clearly define its requirements in this area.

ISE performance requirements -- In all current documentation for ISE there is a significant lack of information regarding NASA's requirements on ISE performance. ISE, as with the rest of the station's application software, will function in an open system of distributed processors supporting multiple users and tasks. It is poor engineering practice to leave important numbers (such as minimum timing response required for finding, and loading command sequences not in active memory) undefined and not testable.

ISE as a managing entity--The management responsibilities of ISE as an executive are confused by the role of managing the TOLs and the secondary power system.

- There is no value added by ISE when controlling TOLs. TOLs are just another set of objects to manage (part of system control).

- There is no value added by ISE when controlling or monitoring secondary power (part of system control).

ISE functions as notification discriminator and a Command Sequencer--The primary functions of ISE are to perform as an integrator of station commands for the SSCC and the POIC. To perform this function, the ISE needs basic function capabilities that are not included in the requirement documents. As was discussed in section 3, these capabilities could be used by many station applications, and as such, they should be commonly available through DMS application services. The basic functions are listed as follows:

- ISE requires a scheduler and Command Sequencer.
- ISE requires a notification discriminator.
- ISE requires a command discriminator that can issue commands or augmented notifications.

The sequencing of commands is covered through requirements placed on executing the OSTP. However, the execution of the OSTP is based on a capability to execute a user's language. The execution of a user's language is not necessary for the execution of a sequence of stored commands. Requirements should be written to include instances of simple sequencers that send stored commands on a schedule. The requirement needs to indicate that the ISE, SSCC, and the POIC need this capability.

The discrimination of notifications and issuing of commands or augmented notifications is a primary of the function of the ISE. It must be able to compare C&W notifications and FDIR notifications to stored patterns and then start the predefined scheduler and Command Sequencer. The requirements need to indicate how the SSCC and the POIC could manage the discriminator(s).

In addition, while the requirements for failure detection, isolation, and recovery (FDIR) are in the NASA requirement documents, no clear set of requirements relates to the test environment and how test management is to be conducted.

Terminology and object definition problems--Conflicts in technical concepts are made worse by inconsistent definitions and terminology. Further, the SSFP has not specified or enforced a way to collect object oriented paradigm information for the program.

From the beginning of MITRE's involvement in this area, we have recognized that the design and development community has been plagued with a limited set of terminology. This problem has been worsened by the almost continual state of redesign and restructuring

of the requirements. Definition of terminology within the program is constantly changing, and the introduction of new terminology has led to many interpretations of the old phrases. Nowhere is this problem more acute than in the subject area of object oriented programming and in the use of its terminology. This document includes appendix A that lists all the terminology discovered during the assessment of ISE.

While the Work Package 2 team is implementing requirements and providing DMS services, the lack of a set of program standards addressing requirements for the management of objects, the management of object state, the management of object relationships, the management of alarm messages, the management of system event notifications, the management of telemetry, the management of access controls, and the management of testing is going to result in a mixed implementation of systems, elements, and payloads. The program needs to adopt a set of standards that will provide an architecture for the operations management of the space station. An adopted set of object management standards could drastically reduce the cost of operations and the sustaining engineering required for *Freedom*.

5.1.2 Recommendations

Requirement clarifications -- modify the ISE and DMS requirements to make them more explicit. Rewrite the requirements to add the following basic functions:

- Add requirements for a Command Sequencer, a Command Discriminator, and a scheduler
- Add requirements for a notification and event discriminator
- Add requirements to establish ISE's performance characteristics
- Add requirements to clarify ISE's resource management responsibilities
- Add requirements to clarify ISE's configuration management responsibilities
- Add requirements to clarify *who* has management responsibilities for journalizing, attribute scan list selection, and telemetry object list selection

Terminology and object definitions -- To help eliminate the confusion problems related to terminology and definitions:

- Establish and publish a **single** SSFP-wide list of definitions
- Adopt as SSFP standards the ISO/IEC International Standards on OSI Management
- Establish as soon as possible a registration authority (configuration management) for SSFP managed objects and data objects

5.2 Trade-offs and Risks in Adopting ISO/IEC Standards

As stated in the subsections of section 4 of this document, the ISO/IEC OSI management standards provide the needed functions and, in some cases, provide more capability than *Freedom* needs. This section lists trades and risks associated with adopting each of the ISO/IEC standards.

5.2.1 Object Management

All of the object management standards ISO/IEC 10164-1 could be adopted by the space program for use onboard and on the ground. However, the object management attribute value change notification type should **NOT** be used for conveying attribute information changes that have either specific notification types defined as a part of the object definitions provided to meet the flight software and object standard, or have TOLs defined to periodically report system, element, and payload attribute values.

The risk in using the object management attribute change notification is that the change could be reported more than one way (If the attribute is time critical or safety related, then this may be desirable, otherwise the double reporting is a waste of communication resources).

The risk of adopting ISO/IEC 10164-1 is minimum. It is an international standard.

5.2.2 State Management

The entire state management model of ISO/IEC IS 10164-2 is applicable to the configuration management of the on-board managed object. This standard model would provide the definitions and terminology needed to clarify the meaning of operational configuration management.

There is one standardized state management notification that reports the standardized state attribute changes of a managed object. The reported changes result through either the internal operation of the managed object (system, element, or payload) or via command operations directed to the managed object (system, element, or payload). If the state attributes are included in the normal space station summarization notifications (TOLs) from the systems and elements, then an event forwarding discriminator should be specified to control the forwarding of state change notifications (see Event Management, section 4.5). The advantages of using the standardized notification are the efficiency of the communications' bandwidth and the control of the notifications using event notification forwarding discriminators. Using standardized notifications removes the reporting of state

variables from the TOLs. The advantage to using TOLs to report state variables is that the SSCC has periodic updates to indicate that the states have **not** changed.

The risk of adopting this ISO/IEC 10164-2 is minimum. It is an international standard.

5.2.3 Attributes Representing Relationship Management

Attributes representing relationships allow managers to control the interactions among two or more systems. Appendix E, section 4, suggests the application of relationship attributes for integration of station modes, system modes, secondary power distribution, and system function inhibits. The relationship attributes discussed in Appendix E.4 are based on International Standards (ISO/IEC 10164-3).

The risk of adopting this ISO/IEC 10164-3 is minimum. It is an international standard.

5.2.4 Alarm Management

There are five ISO standardized alarm notification types used to report the problems of a managed object. The alarms are either the result of an internal behaviour of the managed objects or as the result of management commands. The alarm notification of the alarm function has the parameters discussed in the modeling section 4.4.1 of the alarm management function. The standard specifies the five alarm function notification types as separate alarms reporting notifications. The five specified alarm reporting notifications are as follows:

- CommunicationAlarm
- QualityOfServiceAlarm
- ProcessingAlarm
- EquipmentAlarm
- EnvironmentalAlarm

These five types of alarms along with their severity attributes could be used by Tier 1 components to classify, select testing sequences, or select recovery tactics.

When and if the ISO/IEC IS 10164-4 for C&W alarm classification is used, then agreements on the use of optional fields and subfields of the notifications require careful selection. For example, the optional **proposed repair** action parameter may be more costly than it is worth. MITRE suggests the optional **problem data** parameter be used to report detected "facts" concerning the alarm, and leave the speculation of possible repair actions to operation's controllers or crew.

In addition, the standard provides for six levels of severity. These severity levels are should be mapped to the levels and criticality defined by the on-board safety and time criticalities requirements. Even the ISO/IEC standards for threshold limits, ISO/IEC 10164-5, and security alarms, ISO/IEC 10164-8, only use a fraction of the six levels of severity.

The risk of adopting this ISO/IEC 10164-4 is minimum. It is an international standard.

5.2.5 Event Management

The basic function of the event management is very important to the SSFP.

The ISO/IEC IS 10164-5 specifies the **eventReportRecord** and **eventForwardingDiscriminator** as supporting objects for the event reporting function. The event report record is needed to support the logging of the event reports processed and sent by the event forwarding discriminator object. The event forwarding discriminator becomes the basic filter mechanism upon which ISE operations can be selected. The discriminator construct allows for the examination of C&W notifications and managed object notifications (e.g., FDIR notifications). The adoption of this standard allows operations management of the managed systems (systems, elements, and payloads) in programmable unique ways. A Command Discriminator could also be applied to commands.. (Appendix E provides generic definitions of proposed SSFP managed objects and attributes. One of the managed objects defined is a Command Discriminator that enables Command Sequencers.)

The adoption of the eventReportRecord as a standard would simplify on-board logging of the event reports, since they would be logging a standardized notification.

The risk of adopting this ISO/IEC 10164-5 is minimum. It is an international standard.

5.2.6 Testing Management

This function as specified by ISO/IEC CD 101065-12 is very important to the space station operation. Such a standard would provide the basis for the establishment of failure recovery as directed by the ground controllers and the crew. Testing without such support function could be dangerous and costly.

The disadvantage of applying this standard is that the testing object model may be too complicated resulting in a lot of on-board code. However, the simple testing model based upon built-in test and synchronous testing should be consistent with the SSFP requirements. The more complicated asynchronous testing objects could be added after the station is manned.

The risk of applying the ISO/IEC CD 101065-12 is that it is subject to change before it becomes an international standard.

5.2.7 Log Control Management

Log control is not an ISE function. DMS has the standard service to provide for a uniform log control interface. The standard ISO/IEC 10164-6 specifies the log control function service as a generic storage resource that stores copies of information and is controllable with log control commands. The log is a repository for records. Information to be logged is obtained from reported event notifications, object management notifications, state management change notifications, relationship change notifications, commands, C&W alarm notifications, FDIR alarms, security breach notifications, and communication protocol data units. The log object provides the generic storage resource that is controlled with log control commands.

The advantage of adopting this standard are many. The adoption allows standard management of logs on the ground and onboard.

The risk of adopting this ISO/IEC 10164-6 is minimum. It is an international standard.

5.2.8 Objects and Attributes for Access Control Management

Access control is not an ISE function. DMS has standard services to provide a uniform method for access control using SSFP standard attributes. The objects and attributes for access control management are numerous and complicated. MITRE recommends a selected few access control attributes. The suggested attribute is as follows:

InitiatorADD

Access Control List of Commands associated with particular targets:

- **Operation** attribute
- **ObjectList**

Thus, adopting the ISO/IEC CD 10164-9 would provide more capability than required, however, the application of selected attributes would still be compliant with the standard. Adopting more of the access control attributes could be an expansion after the station becomes manned.

The major risk of adopting ISO/IEC CD 10164-9 is that since it is only a committee draft, it is subject to major change before it becomes an international standard.

5.2.9 Summarization (TOL) Management

Summarization is not an ISE function. DMS has standard service requirements to provide the summarization function. Some of the ISO/IEC CD 10164-11, and ISO/IEC CD 10164-13, could be adopted. The homogeneous and heterogeneous scanner with a numeric attribute list could be used by the SSFP. The advantage of adopting this standard would be the early agreement between ground and the on-board systems as to how telemetry object lists will be formatted and sent to the ground. The establishment of the standard notifications and their parameters would drastically simplify the interface control documents and allow easy modification and adoption of software written to the OSI management standards. The disadvantage would be the use of ASN.1 basic encoding rules, which requires some additional bit overhead.

The risk of adopting ISO/IEC CD 10164-11 and ISO/IEC CD 10164-13 is that they are subject to change before they become international standards.

5.2.10 Scheduling Management

Scheduling management is needed by ISE but should be provided by DMS services. The functions of scheduling an operation (a sequence of commands), summarization notifications (TOLs), testing, communications, and logging are important for the operation of the space station. The advantages of a centralized scheduling function are the reduction of on-board code and the synchronization of operations among two or more managed objects. The disadvantage of adopting such a scheduling function is that if too many managed objects rely on a common scheduling function, then a failure in the scheduling function could result in a large number of timing failures. The best solution may be a limited number of scheduling functions applied to the operations, summarization notifications, testing, communications, and logging.

The risk of adopting ISO/IEC WD 10164-s is that it is subject to change before it becomes an international standard.

LIST OF REFERENCES

NASA, SSP 30555 S1 P1, *Level A Integrated Flight Software Architecture Requirements Document, Section 1: Integrated Avionics, Part 1: Integrated Flight Software Architecture Requirements*, August 15, 1991.

NASA, SSP 30555 S1 P2, *Level A Integrated Flight Software Architecture Requirements Document, Section 1: Integrated Avionics, Part 2: Integrated Station Executive (ISE) System Requirements*, August 15, 1991.

MDSSC SP-M-001, *Contract End Item Specification for the Data Management System, vol. 2, Integrated Station Executive (DR SY-06.1)*, Specification No. SP-M-001, Revision D, September 1991.

Dawson, C.T., Whitelaw, V.A., *Software Restructure Scrub - Summary of Results White Paper*, 3 June 1991, Houston, TX.

NASA, SSP 30261-1, *Architectural Control Document (ACD) Data Management System (DMS), Section I : Integrated Avionics*, 22 March 1991, Reston VA.

NASA, SSP 30261-2, *Architectural Control Document (ACD) Data Management System (DMS), Section I : Integrated Station Executive (ISE)*, 22 March 1991, Reston VA.

NASA, SSP 30000, *Space Station Program Definition and Requirements, Section 3: Space Station Systems Requirements*, Revision K, June 1991, Reston VA.

NASA, JSC 31000, *Space Station Project Description and Requirements Document, Volume 3: Project Design and Development Requirements*, Rev. G, 04 April 1991, Houston, TX

NASA, SP-M-243, *Space Station Program Requirements (Level C), Flight System Software Requirements (FSSR) - Integrated Station Executive (ISE)*, September 1991, Houston, TX.

NASA, SSC-18020, *Software Command and Control Architecture*, September 16, 1991, Reston VA.

ISO 7498, International Organization for Standardization, 1984, *Information technology - Open Systems Interconnection - Basic Reference Model*, ISO/IEC 7498, Secretariat: U.S.A. American National Standards Institute.

ISO 7498-4, International Organization for Standardization, 1989, *Information technology - Open Systems Interconnection - Open System Management Framework*, ISO/IEC 7498-4, Secretariat: U.S.A. American National Standards Institute.

ISO 9595, International Organization for Standardization, 1989, *Information technology - Open Systems Interconnection - Common Management Information Services (CMIS)*, ISO/IEC 9595, Secretariat: U.S.A. American National Standards Institute.

ISO 10040, International Organization for Standardization, 1991, *Information technology - Open Systems Interconnection - Open System Management Overview*, ISO/IEC 10040, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-1, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 1: Object Management Function*, ISO/IEC 10164-1, Secretariat: U.S.A. American National Standards Institute

ISO 10164-2, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 2: State Management Function*, ISO/IEC 10164-2, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-3, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 3: Attributes for Representing Relationships*, ISO/IEC 10164-3, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-4, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 4: Alarm Reporting Function*, ISO/IEC 10164-4, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-5, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 5: Event Reporting Function*, ISO/IEC 10164-5, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-6, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 6: Log Control Function*, ISO/IEC 10164-6, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-7, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 7 - Security Alarm Function*, ISO/IEC 10164-7, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-8, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 8 - Security Audit Trail Function*, ISO/IEC DIS 10164-8, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-9, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 9: Objects and Attributes for Access Control*, ISO/IEC CD 10164-9, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-10, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 10 - Accounting Meter*, CD 10164-10, Secretariat: U.S.A. American National Standards Institute

ISO 10164-11, *International Organization for Standardization, 1991 Information Processing Systems - Open System Interconnection - System Management - Part 11: Workload Monitoring Function*, ISO/IEC DIS 10164-11, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-12, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 12: Test Management Function*, ISO/IEC CD 10164-12, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-13, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part 13: Summarization Function*, ISO/IEC CD 10164-13, Secretariat: U.S.A. American National Standards Institute.

ISO 10164-s, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - System Management - Part s: Scheduling Function*, ISO/IEC WD 10164-s, Secretariat: U.S.A. American National Standards Institute.

ISO 10165-1, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - Structure of Management Information - Part 1 - OSI Management Information Model*, ISO/IEC 10165-1, Secretariat: U.S.A. American National Standards Institute.

ISO 10165-2, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - Structure of Management Information - Part 2 - Definition of Management Information*, ISO/IEC 10165-2, Secretariat: U.S.A. American National Standards Institute.

ISO 10165-4, International Organization for Standardization, 1991, *Information Processing Systems - Open System Interconnection - Structure of Management Information - Part 4 - Guidelines for the Definition of Managed Objects*, ISO/IEC 10165-4, Secretariat: U.S.A. American National Standards Institute.

APPENDIX A

STANDARD TERMINOLOGY AND DEFINITIONS

Abnormal test termination: Abnormal test termination is a statement made with respect to a test invocation when the test is prematurely terminated.

Access control: Access control is defined by ISO-7498-2 as the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner.

Access control certificate (ACC): An access control certificate (ACC) is ACI used to specify the access control parameter used with CMIS. The specification of an access control policy may include the definitions of this ACI. For example, an ACC may contain:

- the identity of the security domain (e.g, the station, a bank) and the security domain authority (e.g., the SSCC, a bank office)
- the ACI required by access control policy. (This information may be one or more of the initiator capabilities [e.g., a system controller], initiator name [e.g., current commander's name] or security labels [e.g., crew member certification], time ACI becomes valid, the time the ACI was created, or the integrity check information [e.g., what command constraints are or are not allowed].)

Access control information: Access control information (ACI) is any information that is used for access control purposes.

Access control list: Access control lists are defined by ISO/IEC 10181-3 as a mechanism that is used by an identity-based access control policy to specify lists of initiators who may or may not have access.

Access control policy: An access control policy is an aspect of the security policy that is specific to access control. For Tier 1, an access control policy specifies the condition in which security services and mechanisms are used to control access to station information. A SSFP access control policy is a coherent set of rules imposed within the security domain of the station by the station security authority. A managed object may be in multiple security domains if some aspects of the managed object are under the jurisdiction of different security policies. When a managed object exists in multiple security domains, the enforced access control policy is the one that corresponds to the policy in which the access request originated. By definition, therefore, the initiator and target in any management exchange are governed by the same access control policy.

Actions: Actions are the commands and/or services that a managed object and its application software can perform. The actions define the possible messages to which the object will respond. Action commands result in object behaviour.

Administrative constraints: Administrative constraints are restrictions on the usage and availability of managed objects. For example, administrative constraints include power status, failure status, test status, preference status, etc.

Alarm report: An alarm report is a specific type of event report used to convey alarm information.

Allomorphism: Allomorphism is the ability of a managed object that is an instance of a given class of managed object to be managed as an instance of one or more other managed object classes. An ordered allomorphic list is a sequence of associated objects related by modifications in behaviour. For example, a revised managed object that may be able to operate as the unrevised managed object is allomorphic.

Asynchronous test: An asynchronous test is a test invocation for which the successful confirmation to the test request does not imply termination of the test invocation.

Capability: (1) Access control capability in terms of the access control framework is defined by ISO/IEC 10181-3 as a mechanism used by a capability-based access control policy to specify lists of capabilities that may or may not have access. (2) the functions of a software application with a specific start and stop time.

Collection period: A collection period is the time during which a metric algorithm is applied to observe data. The collection period includes both the scanning and the calculating times.

Command constraints: Command constraints are sets of command inhibits to restrict and block the listed commands from reaching or affecting a managed object.

Command Enable: The providing of the authority to execute commands by a device or an application.

Command enabling: A command enable provides the access to a device or an application to execute commands. The removal of command inhibits provides command enabling.

Command Inhibit: The temporary removal of the ability to receive commands from a device or software application. An external inhibit blocks the command path to a device. An inhibit may also be the prohibition of command execution placed on a device or software application. An internal inhibit prevents the software from executing commands.

Command inhibiting: (Inactive Capabilities) A command inhibit temporarily removes from a device or software application the ability to send, receive, or execute a command. Inhibiting the ability to send provides access control by preventing the sending of the command. Inhibiting the ability to receive provides access control by a decision function that blocks the command from being delivered to the managed object. Inhibiting execution of a command prevents the behaviour of the managed object (i.e. prevents the behaviour of the system, element, or payload by limiting the activity of the application).

Creation: In the ISO/OSI management standards, the creation of objects means the invoking or loading of the software modules. The ISO use of the word create is similar to the concept of initialization in the DMS ACD. An OSI management creation does NOT imply any God-like behaviour.

Deletion: Deleting managed objects in the ISO standard is the NASA concept of transitioning to the dormant state. In the ISO standards, deletion does NOT imply the destruction of the managed object.

DMS objects: (1) A DMS data object is an abstract representation of the storage spaces for attribute values. DMS data objects can either be read-only or read-write. DMS data objects do not have any operational management behaviour. The behaviour of on-board managed objects are attributed to the hardware characteristics, the environment, and the software applications. The behaviour of the managed objects is the result of changes to the DMS data objects, the environment, and the behaviour rules of the physical and logical managed objects. Examples of DMS data objects are as follows: attributes indicating identity such as attribute names, software application names, and command names; configuration attributes such as enabled or disabled; relationship attributes such as pointers to redundant hardware and pointers to constraints on the modes of systems. (2) DMS STSV objects do have behaviour and react to commands. For example the DMS message object, and DMS telemetry object list object send predetermined notifications.

Functional objects: Functional objects are software applications that have defined capabilities (i.e. behavior that is the result of either changing attribute values, commanding actions to change the process, changing the object's environment, or the behavior rules associated with the object. Examples of functional objects are software applications included in the systems, elements, payloads, orbital replaceable units, standard data processors, mass storage units, MDMs, etc.

Granularity period: Granularity period as defined in ISO/IEC 10164-11 is the time between observations of a managed object. In the context of ISO/IEC 10164-13, granularity is the time between the initiation of two successive scans.

Heterogeneous scanner: A heterogeneous scanner is a scanner that collects values from potentially different sets of attributes for a set of explicitly named observed object instances and reports the results at the end of each scan.

Homogeneous scanner: A homogeneous scanner is a scanner that collects values from the attributes of the same type from one or more managed objects. (For example, the scanning of temperature values from temperature sensors.)

Initiator: An initiator is the entity that makes the management request for action or information.

Inhibit: See command inhibiting

Interlock: An interlock managed object is an extra hardware or software control used to ensure additional operational safety. Interlocks provide functionally independent control over a process or a device that may be hazardous or disruptive. Interlocks have behaviour that require a two-step command sequence.

Intermediate Language Executer (ILE): Specific use of this DMS service and its existence in the program are the subject of debate at this time. *Level A* requirements here must be considered subject to change. For a detailed description of the ILE and UIL see SSFP UIL Specification, USE 1001 15 March 1990.

Interval scheduling: Interval scheduling is a type of scheduling that controls a number of intervals of operation of activities within specified managed object instances.

Locked DMS objects: The DMS object lock prevents user's access to the replacement of RODB values. The current command authority continues to have access to the replacement of the RODB values.

Locked managed objects: The administrative lock prevents user's access to the behaviour of managed objects. The administration authority continues to have access to the behaviour of the managed objects. SSFP administrative authority consists of any authorized source. The possible administrative authorized sources include crew, SSCC ground controllers, POIC controllers, and international partner personnel.

Log record: A log record is a management support object class that models units of information stored in a log.

Log: A log is a management support object class that models resources used as a repository for log records.

Managed object under test (MOT): The managed object under test (MOT) is the managed object that represents a management view of the resource or resources whose function is the subject of the test.

Managed objects: A managed object is an abstract representation of resources of a managed system. The management of these resources require a management view of the logical and physical identities within the managed system. Managed objects have behaviour and the managed object behaviour is the result of either changing attribute values, commanding actions to change the managed process, changing the managed objects environment, or the behaviour rules associated with the managed object. Examples of managed objects are systems, elements, payloads, orbital replaceable units, standard data processors, mass storage units, etc.

Management: Management of the *Freedom* objects is the commanding and monitoring of the *Freedom's* systems, elements, or payloads. The RODB and IODB contains management views of the attributes of all the managed objects.

Media data: In this document, media data is data about data. Media access control data is the attribute values of attributes associated with access control object, e.g., security policy rules, authorized list of controllers and commanders, list of actions related to transitioning between station modes, etc.

Metric attribute: A metric attribute is defined in ISO/IEC 10164-11 as an attribute if a metric object whose value is either used as a parameter of one or more metric algorithms or whose value represents the output of such an algorithm.

Metric object: A metric object is defined in ISO/IEC 10164-11 as a managed object that contains at least one attribute whose value is calculated from the values of attributes observed in managed objects.

Numeric attribute: A numeric attribute is an attribute whose value may be either an integer (or possibly treated as an integer) or real number.

Object: (1) A thing that can be seen, touched, or can logically exist and has attributes and responses to actions. See DMS object. (2) An abstract representation of a physical or logical entity. An object is characterized by its attributes, notifications, actions, and behavior. See managed objects, test objects, metric objects, and functional objects.

Observed attribute: An observed attribute is an attribute of a managed object, system, element, or payload whose value is being observed by a metric object or a summarization object. (For example, all attributes in the RODB are observable. Any derived values from the attribute values in the RODB are provided by a metric object. The DMS

STSVs that scans the RODB and generates telemetry object lists are summarization objects.).

Observed object: An observed object is a managed object with attribute values that are observed by a metric object or a summarization object.

Optional parameters of notifications: Optional parameters of notifications are fields of the report that may be included in the notification services if the user chooses. Mandatory parameters must be in the notifications.

Orbit-time: Is the period of time associated with a few orbits. This term is used in preference to *real-time* because the context of real-time implies a fast reaction time.

Package: An OSI package is a set of optional behaviours with associated attributes that are imported into the managed object at the time the managed object was created or initialized. A conditional package is one of a set of packages that is selected based upon conditional attributes supplied at the time the managed object was created or initialized.

Parameter of a notification: A parameter of a notification is the reported bit field that is to be filled with an attribute value. The coding of the attribute value in the parameter is to follow the standardized ASN.1 transfer syntax as specified in the ISO/IEC IS 10165-2.

Parameters of an action: Parameters of an action are the bit fields of the command.

Periodic scheduling: This is a type of scheduling that controls the repetitive triggering of activities with specified managed object instances.

Prohibits: The administrative permissions and prohibits are controls used to stop commands or processes while administration (the crew or the SSCC ground controllers) monitor the attributes of the managed objects and control the managed objects. The administrative permission and prohibits are effectively locks on the use of the managed resources.

Protected entity: A protected entity is a set of one or more items that are protected by the same access restrictions. An item may be a software application process, a set of attributes controlled by an application process, or a set of values of a single attribute controlled by an application process.

Registration: Registration is the process of defining and putting under configuration management control the definitions and transfer syntax of attributes, objects, notifications (and TOLs), actions (commands), packages, behaviours, and relationships. The ISO/IEC standard 10165 specifies how to complete the template (forms) necessary

to "freeze" a managed object. Registration involves some authority (e.g., NASA configuration management change boards) to publish and duplicate the information for others to use. The registration of an object makes it an object with defined interfaces and properties with which others can communicate and obtain services.

Relationship: An OSI relationship is a set of rules that describe how the operation of one part of a system, element, or payload affects the operation of other parts. A relationship is said to exist among managed objects when the operations of one managed object affects the operation of the other managed objects. For a relationship to be significant within the context of managing the operation of *Freedom*, sufficient information must be available to allow Tier 1 to identify the managed objects involved and the rules governing their interaction.

Report period: The report period is the time between emitting notifications containing the collected aggregate values or statistical information.

Resource constraints: Resource constraints are attribute value limits associated with managed objects that represent consumable resources. For example, most managed objects have power consumed and heat generated attributes. Depending on the operational states and power states of the managed objects and the number of managed objects (system, element, or payload), the sums of the power consumed and heat generated will determine the total power consumed and heat generated. The limits on the sums or on the individual attribute values are resource constraints.

Scan: A scan is a sampling process of observing attribute values at a specified point in time.

Scheduled managed object (SMO): The scheduled managed object is the managed object whose activities are to be scheduled.

Scheduler object (SO): A scheduler object is the managed object that defines the type and values of the schedule to be applied to activities within the SMOs.

Scheduling function: The scheduling function is the method of controlling the timing of the performance of a scheduled activity which are represented by a managed object.

Security label: A security label in terms of the access control framework is defined by ISO/IEC 10181-3 as a mechanism used by a label-based access control policy to specify lists of security clearances that may or may not have access.

Security policy: Security policy is defined in ISO 7498-2 as the set of criteria for the provisions of security services.

Signature: A signature is defined by ISO 7498-2 as data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protects against forgery (e.g., by the recipient).

Specified alarm: An ISO/IEC specified alarm is a notification of the form defined by the ISO/IEC IS 10156-4 alarm function and a specific event. An alarm may or may not represent an error.

Subparameter: A subparameter of a notification parameter is the reported included bit subfield that is to be filled with an attribute value. The coding of the attribute value in the subparameter is to follow the standardized ASN.1 transfer syntax as specified in the ISO/IEC IS 10165-2.

Summarization: A summarization is the process of aggregating and optionally applying algorithms to obtain raw or observable information to produce summary information.

Support Object: A support object is an abstract representation of logical and functional element managers that aid in controlling other objects. For example, TOL objects, scanners, sequencers, discriminators, and schedulers are all management support objects.

Synchronous test: A synchronous test is a test invocation for which any confirmation to the test request implies termination of the test invocation.

Systems, elements, and payloads: The systems, elements, and payloads are managed objects that contain other managed objects. The states of the system, elements, and payload managed objects follow the same definitions that apply to the states of their contained objects. The collection of the state attributes values of the contained objects provide a detailed view of the states of the contained objects in the system, element, or payload.

Target: A target is the entity to which the access request is addressed. For example, targets are the objects, systems, elements, and payloads that are commanded.

Template: Templates are the defined formats or forms for various object oriented properties. The ISO/IEC 10165-4 standard defines the templates for notification, objects, attributes, behaviour, etc. Thus, ISO/IEC 10165-4 tells object definers how to "spell" and how to "write and generate standard information" about the various properties of objects.

Test: A test is the operation and monitoring of an object, system, element, or payload within an environment designed to elicit information on the functionality and/or the performance of the subject.

Test action request receiver (TARR): The test action request receiver (TARR) is a term use to identify the ability of a managed object to act upon a test request. For example, the SMI is a TARR since it can receive test commands and invoke test sequences.

Test conductor: A test conductor is a manager or managing object that issues test operations (i.e., commands).

Test invocation: A test invocation is a specific instance of test, from the time of initiation to termination.

Test performer: A test performer is an agent (i. e., an object or system) that receives test operations (i.e., commands).

Test session: A test session is a set of test invocations.

Testing object (TO): A testing object (TO) is a managed object that exists only for the duration of a test invocation and which has attributes, behaviours, and event notifications that pertain to that instance of test. It issues the specific test, measures the test responses, and generates the resulting test event notifications.

Threshold: An ISO threshold is modeled as an attribute with two levels: the triggering level and the clear level. Each of these threshold levels may be triggered on either an increasing or decreasing gauge values. By setting the attributes of the two threshold levels and values of the decreasing and increasing attributes, an ISO threshold can be functionally set to have a specified hysteresis (the difference between the triggering level and the clear level) and trigger or clear only on the first crossing in an increasing or decreasing direction. ISO/IEC IS 10165-2 specifies these threshold attributes and their values. ISO/IEC CD 10164-11, clause 7, provides the generic threshold model description.

APPENDIX B

OPEN SYSTEMS MANAGEMENT TUTORIAL

The basic abstract model, or framework, for OSI management is as illustrated in figure 12. The basic framework consists of a manager system and one or more systems to be managed. Management is treated as a distributed application with components residing within a manager system (managing processes) and managed systems (agent processes).

Management activities are affected by managing processes communicating with remote agent processes to manipulate managed objects contained in the managed systems. Defined for each managed object are:

- Class of object (e.g., sensor, effector, interlock, system module, application software module, etc.)
- Configuration states (e.g., usage states: object enabled, object disabled, object idle, object busy; administrative states: locked, unlocked; availability status: failed, initiated, dormant, reported in a FDIR message, etc.)
- Relationships (back-up and backed-up, primary and secondary, system mode component, powered standby and unpowered standby, contained within, where-used, etc.)
- Attributes (e.g., counters, thresholds, resource constraints, operating limits, access control, etc.)
- Valid operations (e.g., commands, command constraints)
- Behaviour (e.g., the actions the object does as the result of the data values of the attributes and the commands received)
- Notifications (e.g., FDIR messages, test messages, behaviour messages, initialization messages, deactivation messages, attribute change messages)

A definition for each of the above means that the actual bit pattern fields representing the characteristic is structured and its syntax of values is determined. For example, the definition of the notifications include the identification of parameters (message fields) and subparameter (message subfields) as well as how the values will be represented in the

message. Both manager and agent processes must have the same shared conceptual view of the managed objects in terms of attributes, relationships, behaviour, operations, and notifications that may be emitted. A managing process may monitor or control one or more agent processes. An agent process manages the associated managed objects. That is, an agent process performs management operations such as reading or modifying attributes of a specific object or set of objects, as requested by a managing process, and may return a response to the managing process. Agent processes may also forward notifications (events) asynchronously generated by managed objects. Agent processes may have supporting services as follows:

- Summarization notifications (telemetry) of a management selected set of the defined and measurable attributes
- Scheduling of periodic notifications or behaviour
- Logging of records of commands, notifications, test results, C&W alarms
- Setting (writing) of attribute values
- Getting (reading) of attribute values
- Reporting of alarms that exceed thresholds (C&W alarm messages)
- Testing and reporting notifications of the availability status of the object (FDIR services)
- Security alarm reporting (security breach messaging)
- Discriminators and filters that limit the message content and message address.

As illustrated by the basic management framework, figure 12, SSFP management functions must include definitions of the data being managed, the operations that can be performed on these data, and the protocols supporting the sending of these operations across the networks. Thus, the management tools most essential to standardize for the ISE include:

- A common framework and terminology for describing management concepts in terms of the logical components that take part in management, and an application of that model to the layered protocol architecture

- A set of communication services and related protocols to transfer management operations and responses as well as notifications between managing and agent processes
- A common structure for the management information (The structure of how the RODB *spells* in bits the configuration state values, the relationships, attribute values, the commands, the notifications, and the behaviours)
- Detailed specifications of the managed objects and their attributes, the management operations permitted on them, notifications they may emit, and how the definition of the managed object relates to the behaviour of a real entity it describes

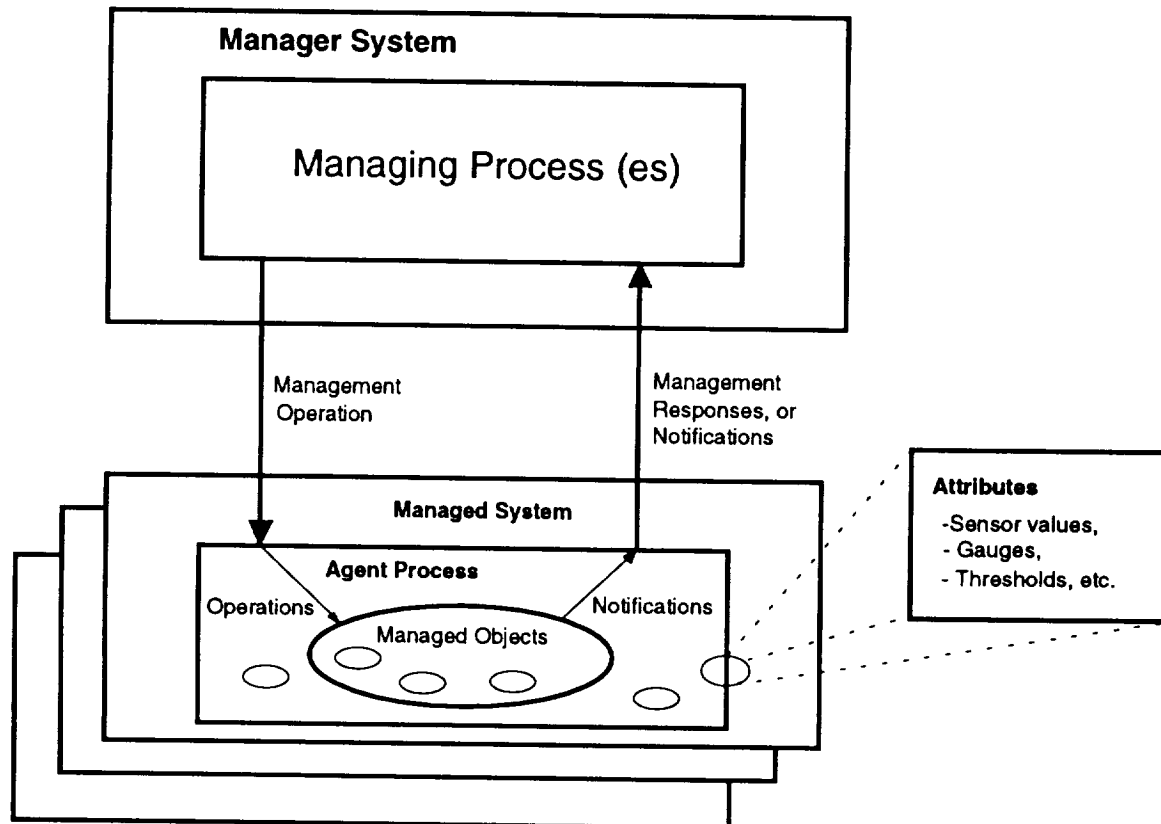


Figure 12. Basic Management Framework

The current developing OSI standards provide the basis for structuring the SSCC network management function. Section 4 presents the current OSI international standards providing for Common Management Information Services and a Common Management Information

Protocol for controlling the managed process, and for OSI management of each SSFP object, system, element, or payload. The current international standards provide for the following open system interconnection functions:

- Object management (e.g., administrative, operational and usability controls)(ISO, 1991 [10164-1])
- State management (e.g., enabled, active, busy, disabled, locked, unlocked, failed, reported faulty)(IS, 1991 [10164-2])
- Relationship management (e.g., organization and operational dependencies between supporting agents)(ISO, 1991 [10164-3])
- Notification management (alarm reporting function (ISO, 1991 [10164-4]) and event report management function (ISO, 1991 [10164-5])
- Log control management (ISO, 1991 [10164-6])
- Security alarm management (ISO, 1991 [10164-7])

In addition, current international standards present the OSI Systems Management Overview, the OSI Management Information Model (ISO, 1991 [10165-1]), the OSI Definition of Management Information: (DMI) (ISO,1991 [10165-2])), and the OSI Guidelines for the Definition of Managed Objects: (GDMO) (ISO,1991 [10165-4]). These ISO standards include many defined objects and attributes to be used in the open system interconnection environment.

In addition, current draft international standards and committee drafts provide information for the standardization of the following:

- Security Audit Trail (ISO, 1991 [DIS 10164-8])
- Objects and Attributes for Access Control (ISO, 1991 [CD 10164-9])
- Accounting Metering (ISO, 1991 [CD 10164-10])
- Workload Monitoring Function(ISO, 1991 [DIS 10164-11])
- Test Management Function (ISO, 1991 [CD 10164-12])
- Summarization Function (ISO, 1991 [CD 10164-13])

A current OSI management working document that provides guidelines for a scheduling function is (ISO, 1991 [WD 10164-s]).

APPENDIX C

MANAGEMENT SERVICE CONTROL STANDARDS

ISE Management Service Control Function

As illustrated in figures 1 and 2, the ISE manages its functions following a object oriented model. The International Organization for Standardization and the International Electrotechnical Commission, Joint Technical Committee, Sub Committee 21, Working Group 4, has recently completed the recommendation of the standardization of a set of management functions, managed objects, and the object attributes consistent with an OSI management object oriented model. These standards are contained in ISO/IEC 7498-1, 7498-4 and 10164 -1 through 10164-7. The same committee is working on standards ISO/IEC 10164-8 through 10164-13.

The discussion presented below provides a description of the ISE functions provided by a management control function and the management information used by the ISE functions. This functional description meets the requirements of the SSFP Tier 1 to monitor and control command constraints, object resource constraints, object behaviour, and the station modes.

The SSFP Tier 1 also needs a consistent set of definitions and actions related to the management of the commands that *Freedom's* systems, elements, and payloads receive. In cooperation with the *Freedom* object management function (see section 4.1) and state management function (see section 4.2), Tier 1 needs the ability to implement management service controls such as command constraints and overrides.

Management Service Control Model

Each *Freedom* object is subject to management service control. The managed objects and their attributes are to be defined in accordance with appendix D, the Flight Software Data and Object Standard, of the DMS ACD SSP 30261. This data standard refers to an applicable document, the SMI, ISO/IEC 10165. SMI part 2 uses the ISO functions described in ISO/ICE 10164. The Management Service Control function for the ISE has been engineered to be a combination of the ISO/IEC IS 10164-3, *Information Processing Systems - Open System Interconnection - System Management - Part 3: Attributes for representing Relationships* and ISO/IEC DIS 10164-9, *Information Processing Systems - Open System Interconnection - System Management - Part 9: Objects and Attributes for Access Control*. These specifications provide detail on the management of relationships of the management service control function, and they provide a consistent set of definitions that comply with the *Basic Reference Model* (ISO/IEC 7498), the *Open System Management Framework*

(ISO/IEC 7498-4), the *Common Management Information Services CMIS* (ISO/IEC 9595), and the *Open System Management Overview* (ISO/IEC 10040).

The ISE management service control function needs to provide the following:

- Command inhibits and command constraints (sets of command inhibits)
- Overrides of the command constraints and command inhibits
- Overrides of the object behaviour by interactively forcing the execution of special object behaviour sequences
- Overrides of the managed object attributes that limit or select object behaviour

Figure 13 illustrates the proposed management service control model. Command enables reside as part of the attributes associated with the managed objects (see appendix E.4.2 - The Freedom Managed Object Common Attribute Model). The other access control information includes authorized sources and data object locks. These access controls include the capability to examine all commands to a managed object and to block or prohibit their execution by the managed objects. The initialization and instantiations of a set of command enables could be related to a station mode relationship attribute. One attribute of the command enable list would be a command constraint override. In the event a DMS RODB WRITE is used to set the override attribute value to object enabled, then the command constraint list would not be used.

Figure 13 also illustrates the management service control model as it relates to interactive control of an object within a managed object (system, element, or payload). In the object model presented in the ISO standards, each managed object has its defined object behaviour. The behaviour of each object is determined by data attributes. In other words, the object's action (behaviour) is defined by its object class. Objects with different behaviours are in different classes. In order to obtain interactive object behaviour override capability and comply with the standards, then either object actions need to be selectable by individual actions (or a general action with a selection parameter), or the atomic behaviours need to be associated with unique atomic object instance each with an object identifier. If the individual actions are selectable, then they can be related using the group relationship of the ISO standard 10164-3. If the individual atomic behaviour is associated with unique atomic object instances, then the member relationship and owner relationship can be used to represent the contained relationship that the individual atomic object instances have with the owner object. The ISO/ IEC 10164-3 relationship change notification provides a means to report member and owner relationship changes, as well as, group relationship changes. See section 4.3.

Figure 13 also illustrates that normal commanding of object attributes can be used to change attribute values. This provides the capability to override any replaceable data attribute. The change attribute notifications and TOLs (summarization objects) report the changed and current values of the changed attributes.

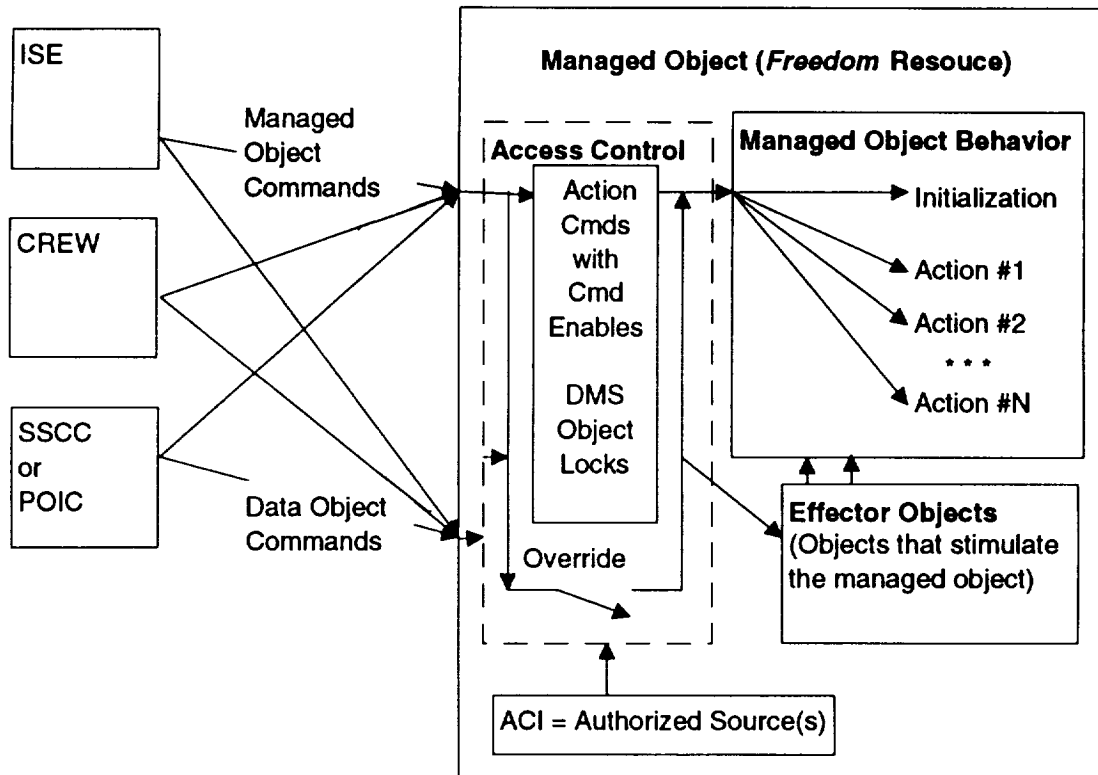


Figure 13. The Management Service Control Function

Management Service Control Change Notifications

The management of service controls is provided by the use of the managed object management function notifications, the state management function change notification, the relationship function change notification, and the notifications of the ISO access control objects and attribute standard. The management service control information could also be included in TOLs from the systems and elements.

Management Service Control Service Definitions

The detail design of the DMS will provide the management service control function. The DMS should have standard services to provide the services listed in the management service

control model and management service control notification sections. The design of the DMS does not have to comply with ISO/IEC IS 10164, but the functionality of the management service control will require equivalent services and information. Appendix E proposes managed objects and common attributes for the *Freedom* managed object classes and includes the definition of attributes to support the access control of the Space Station.

The DMS provides a standard service called "Action I/O." This service retains access control information, provides the access control decision function, and enforces access enforcement information as it relates to operational management commands. In the following paragraphs, the functioning of Action I/O is explained as to how it relates to the inhibits on the input actions to applications, and inhibits on the outputs of applications.

How this access control model might be applied to the onboard software to provide an inhibit capability is shown in figure 14. Any application with access to the DMS STSVs (via the APID) can initiate an Action I/O request. Action I/O receives an object/action pair and determines through the Action Reader Directory (ARD) whether an application has registered to receive this object/action pair. The ARD maintains a dynamic directory of object/action pairs that applications have registered to receive by opening `read_action_handles`. If ARD finds the object/action pair in the directory, then the Action Interface Services of the RODB Executive (REX) forwards the action to the object. If the object/action is not found in the directory, the action is not forwarded and an error response is returned to the initiator.

In summary, the DMS Action I/O service performs both the access enforcement and the access decision functions of the ISO model for all applications using the APID. Further, DMS Action I/O uses an object/action directory (the ARD) to determine what to do with a given action request. The object/action directory is an implementation of ISO Access Control Information (ACI) for target objects. The only feature of the ISO model for access control missing from the current design is the ability to externally manage the Access Control Information.

It must be understood that the ISO model for access control is only a part of a set of ISO standards that define the structure of management information (SMI) necessary for the control of open systems. The ISO SMI assumes a management view of an object, so that if an entity wants to control an object, it must be provided "visibility" to that object. From the SSFP perspective, this means applications (systems) must provide Tier I the ability to access the function it is required to control; i.e., the object action/pair (or set of object/action pairs that together provide a function).

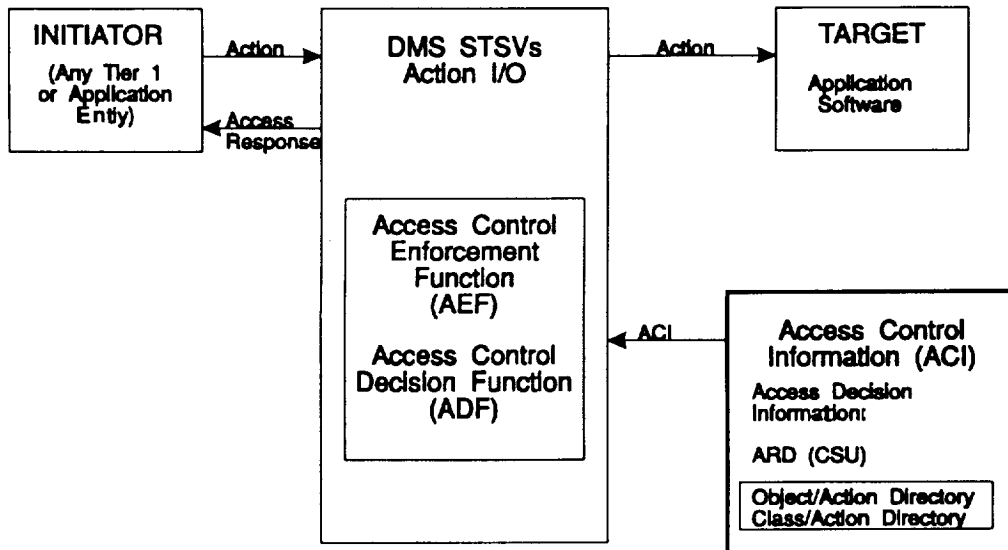


Figure 14. ISO Access Control via DMS STSVs - Action I/O

Adding the capability to the applications to externally manage the DMS object/action directory would make the interaction between applications and Action I/O compatible with the ISO guidance. More importantly, however, it would also provide Tier 1 a management view and external control over each registered object/action pair. This means it could support (or manage) the required inhibit capability for all commands.

Another advantage of making Action I/O compatible with the ISO model to implement inhibits is that the same ISO model supports requirements that DMS Services must provide in future releases. In particular, in a future release DMS Services must support access authorization requirements. In order to support this requirement using an implementation compatible with the ISO model for access control, only the definition or attributes of some of the objects used in the ACI must be changed to accommodate the requirement. Little or no changes would be required in the AEF or ADF software. It is noted however, that the DMS release 2 design for Action I/O is tightly coupled to the structure of the ARD and does not support separate enforcement and decision functions. It is suspected therefore, that significant modification to DMS services (Action I/O) will be required to support access authorization.

Management Service Control Protocol and Abstract Syntax Definitions

The management service controls parameter syntax is defined by the other management functions. The ISO/IEC standard, 10165-2 defines abstract syntax for the standard attributes.

APPENDIX D

EXAMPLE OF GDMO

This appendix includes examples of two managed objects written to the template of the Guidelines for Definition of Managed Objects (GDMO) (ISO, 1991 [10165-4]). The objects are called the scanner and the heterogeneous scanner. The GDMO attributes, actions (commands), behaviour, and notifications characterize these objects. This example is of a summarizer. This managed object is similar to the functions required for telemetry in the SSFP. The DMS data objects are the attributes of the scanner managed object. The notification of scanner managed object is a telemetry packet. The list of attributes scanned and reported in the telemetry packet is the DMS telemetry object list (data) object (TOLO).

scanner MANAGED OBJECT CLASS

 DERIVED FROM top;

CHARACTERIZED BY

 scannerPackage PACKAGE

 BEHAVIOUR DEFINITIONS

 scannerBehaviour BEHAVIOUR

 DEFINED AS

 When the scheduling causes the scanner to become active and granularity period is non-zero, a scan is initiated. The scanner continues to scan at the end of each granularity period as long as the scanner is active according to the schedule. ;

 ATTRIBUTES scannerID GET,

 administrativeState GET-REPLACE,

 granularityPeriod GET-REPLACE,

 availabilityStatus PERMITTED VALUES Attribute-

ASN1Module.DiscriminatorAvailability GET,

 operationalState GET;;;

CONDITIONAL PACKAGES

 dailyScheduling PRESENT IF both the daily scheduling package and external scheduler packages are not present in an instance,

 weeklyScheduling PRESENT IF both the daily scheduling package and external scheduler

 externalScheduler PRESENT IF packages are not present in an instance, both the daily scheduling package and weekly

scheduling packages are not present in an instance;
-- see clause 7 in CCITT Rec.X.734 | ISO/IEC 10164-5 for the description of these packages.
REGISTERED AS { smi2MObjectClass x };

heterogeneousScanner MANAGED OBJECT CLASS
DERIVED FROM scanner;
CHARACTERIZED BY

heterogScannerPackage PACKAGE
BEHAVIOUR DEFINITIONS

heterogScanner Behaviour BEHAVIOUR DEFINED AS

An observation is made of all attributes in observationIDList. A scan that is initiated during an active scheduled period shall be completed normally and the summarization notification emitted. The heterogeneous scanner reports the scanned value at the end of the granularityPeriod for all attributes in the scanAttributeIDList and reports an array of sequential numeric values for attributes in the numericAttributeIDArray. A scan that is initiated during an active scheduled period shall be completed normally and the summarization notification emitted;

ATTRIBUTES

observationIDList GET-REPLACE ADD-REMOVE;
-- list of instances, with one or more attributeIDList for each observed object instance.
-- One type of attributeIDList is for basic scanning of attribute values.
-- Another type of attributeIDList is for scanning and reporting numeric attribute values.

NOTIFICATIONS

heterogenousScanReport;

ACTIONS

activateScanReport;;

REGISTERED AS { smi2MObjectClass x };

dailyReportSchedulingPackage PACKAGE
BEHAVIOUR DEFINED AS

This conditional package provides for the capability of scheduling reporting with a periodicity of 24 hours. It defines a list of time intervals (interval-start and interval-end times of day).;

ATTRIBUTES

reportIntervalsOfDay DEFAULT VALUE Attribute-ASN1
Module.defaultIntervalsOf Day
GET ADD-REMOVE;
REGISTERED AS {smi2Package x};

externalReportSchedulingPackage PACKAGE

BEHAVIOUR DEFINED AS

This conditional package provides for the capability of scheduling reporting based on a schedule defined in an external scheduler managed object;

ATTRIBUTES

reportSchedulerName GET-REPLACE;
REGISTERED AS {smi2Package x};

weeklyReportSchedulingPackage PACKAGE

BEHAVIOUR DEFINED AS

This conditional package provides for the capability of scheduling reporting with aperiodicity of one week. It defines a list of time intervals (interval-start and interval-end times of day).;

ATTRIBUTES

reportStartTime GET-REPLACE;
reportStop Time DEFAULT VALUE Attribute-ASN1Module.defaultStopTime GET-
REPLACE,
reportWeekMask DEFAULT VALUE Attribute-ASN1 Module.defaultWeek
Mask GET ADD-REMOVE;
REGISTERED AS {smi2Package x};

ATTRIBUTE DEFINITIONS:

observationIDList ATTRIBUTE

WITH ATTRIBUTE SYNTAX SummarizationASN1Productions.ObservationIDList;
MATCHES FOR Equality Set Comparison Set Intersection;
REGISTERED AS {smi2AttributeID x};

NOTIFICATIONS:

heterogenousScanReport NOTIFICATION

BEHAVIOUR heterogenousScanReportBehaviour;

MODE CONFIRMED AND NON-CONFIRMED;

ACTIONS:

activateScanReport ACTION
 BEHAVIOUR activateScanReportBehaviour;
 MODE CONFIRMED AND NON-CONFIRMED;
REGISTERED AS {smi2Action x};

BEHAVIOURS:

activateScanReportBehaviour
BEHAVIOUR DEFINED AS
This action type is used to stimulate a scanner object to emit a non-scheduled report.

heterogenousScanReportBehaviour
BEHAVIOUR DEFINED AS
This notification type is used to report that a heterogeneous scan reporting event has occurred.

This clause specifies the heterogeneous scan report, and maps it onto the CMIS M-EVENT-REPORT service.

Table D.1 Heterogeneous Scan Report Notification

Summarization Parameter Name	Corresponding CMIS Parameter Name	Req /Ind	Rsp /Cnf
Invoke Identifier	Invoke identifier	M	M=
Mode	Mode	M	-
Summarization Object Class	Managed object class	M	-
Summarization Object Instance	Managed object instance	M	-
Heterogeneous Scan Report	Event type	M	C=
End Time	Event time	M	-
Observation Scan List:			
Observed Object Instance	Event information	M	-
Time Stamped Attribute List	Event information	M	-
Time Stamped Numeric Value	Event information	U	-
Array	Event information	U	-
Current Time	Current Time	-	U
Errors	Errors	-	C

Summary Notification Discrimination Parameters:

The following list of parameters in a heterogeneous scan report notification needs to be available for use in discriminator constructs: Managed Object Class, Managed Instance, Event Type, and Event Time.

SummarizationASN1Productions {joint-iso-ccitt ms(9) smf(13) asn1Module(??) 0}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

--EXPORTS everything

IMPORTS

WeekMask, ObservedValue FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2)asn1Module(2) 1}

AttributeID, ObjectInstance, BasedManagedObjectID, Attribute, Scope FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) version1(1) protocol(3)};

AttributeMeas ::=SEQUENCE{
attributeID AttributeID,
attributeValue ANY DEFINED BY attributeID,
timeStamp [3] Generalized Time OPTIONAL,
suspectFlag [4] BOOLEAN DEFAULT FALSE

HeterogeneousScanReportInfo ::= ObservationScanList

HomogeneousScanReportInfo ::= SET OF Scans

NumericAttributeIDArray ::= SEQUENCE OF AttributeID

ObservationIDList ::= SET OF ObservationID

ObservationID ::=SEQUENCE{
COMPONENTS OF BaseManagedObjectID,
scanAttributeIDList SET OF AttributeID OPTIONAL,
-- report values at end of each granularity period numericAttributeIDArray
SEQUENCE OF AttributeID OPTIONAL
-- array of numeric attributes whose values are to be placed into buffer numeric value
array

```

ObservationScan ::= SEQUENCE {
    objectInstance ObjectInstance,
    timeStamped AttributeList [0] SET OF AttributeMeas OPTIONAL,
    timeStampedNumericValueArray [1] TimeStampedNumericValueArray OPTIONAL
}

```

--The observed object summary is the combined output from several observed objects, each having its own lists of observed attribute outputs.

```

ObservationScanList ::= SET OF ObservationScan

```

```

PerfTimeInterval ::= CHOICE {
    seconds [0] INTEGER,
    minutes [1] INTEGER,
    hours    [2] INTEGER,
    days     [3] INTEGER
}

```

```

ScanAttributeIDList ::= SET OF AttributeID -- import from CMIP

```

```

TimeOffset ::= PerfTimeInterval

```

```

TimeStampedNumericValueArray ::= SEQUENCE OF -- ordered by time
    SEQUENCE {
        value ObservedValue,
        timeStamp GeneralizedTime OPTIONAL,
        suspectFlag BOOLEAN DEFAULT FALSE
    }

```

APPENDIX E

PROPOSED COMMAND SEQUENCER AND DISCRIMINATOR OBJECT CLASS DEFINITIONS

This appendix presents two proposed definitions of management support⁴⁶ object classes for use by ISE and other station application software. Definitions are presented for a Command Discriminator object class and a Command Sequencer object class. Each object class definition includes a discussion of the need for the object class and the SSFP requirements it will support. The discussion provides a model for each object class and then identifies and defines each of the object's attributes, its behaviour and the acceptable actions or commands. Finally, each object class definition is described in detail using both the GDMO template of ISO/IEC 10165 and the DMS IRD object template.

E.1 Proposed ISE Managed Objects

E.1.2 The Need for a Command Sequencer and Command Discriminator

The Command Discriminator and the Command Sequencer support managed objects provide the needed ISE functions to perform remote operations management. It is important to implement the ISE functions in software applications of support managed objects for several reasons.

- The normal command interfaces both onboard and on the ground can be used to control command sequence execution and discriminator execution without modification or extensions.
- Over the life of the Space Station Program, additional command sequence instances and Command Discriminators can be created whenever necessary as part of the normal on-board reconfiguration process.
- The use of support managed objects fits within the space station standards and results in encapsulation of the managed objects so that they can be coordinated and arranged to perform new operations management functions.

⁴⁶ Support Object: A support object is an abstract representation of logical and functional element managers that aid in controlling other objects. For example, TOL objects, scanners, sequencers, discriminators, and schedulers are all management support objects.

- The application of "standard" Command Discriminators and Command Sequencers would allow all core systems and payloads the functions of these managed objects
- A user friendly interface language could be added to expedite the use of the Command Discriminators and Command Sequencers. Languages such as the TIMELINER and User Interface Language (UIL) or some other computer aided software could be taught to crew and ground controllers to make operations management of the Space Station safer and more cost effective.
- The Command Sequencer and Command Discriminator correspond to the basic functions of programming. They are the building blocks upon which the operations management could be constructed.

Figure 15 illustrates two structures provided by the Command Sequencer and Command Discriminator. The structures illustrated are the:

- DO x AND IF Assertion=TRUE, THEN DO y and the
- WHENEVER Assertion=TRUE, THEN DO z logic constructs.

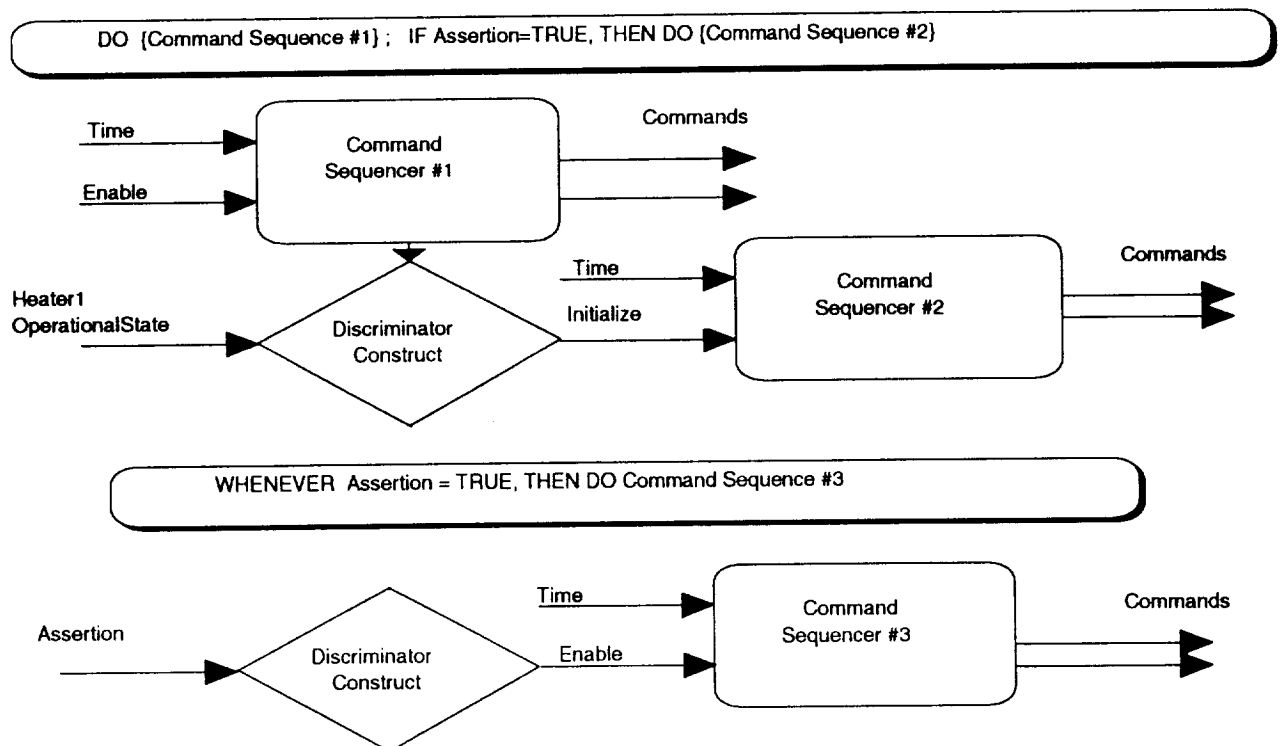


Figure 15. The Simple Programming Constructs

Other structures, such as:

- DO w WHILE Assertion=TRUE,
- DO x UNTIL Assertion=TRUE,
- IF Assertion=TRUE, THEN y ELSE (If NOT Assertion) DO z

can be assembled by combining the instances of the Command Discriminator and the Command Sequencer. With these two objects, simple to complex operational management operations can be provided. Furthermore, these basic objects become the building block upon which user interface languages and computer aids could be added. This approach also allows the reuse of code.

E.1.3 An Example of a Simple *Freedom* Command Sequence

The following example illustrates how a Command Sequencer and Command Discriminator could be used to perform a simple operations management procedure. James Carvajal, NASA-ER2 and Richard Lehman, LESC, wrote a paper on the use of command sequence constructs as a user interface language (UIL) replacement for the on-board Integration Station Executive (ISE). This paper included a simple sequence that provided a hypothetical procedure that used PROCEDURE, RETURN, ON_ERROR, WAIT, PERFORM, VERIFY constructs. That sample sequence is repeated in table 4. How this sample sequence could be performed by a set of Command Sequencers and Command Discriminator is illustrated in tables 5 - 9. Figure 16 illustrates that two sequencers, and three Command Discriminators perform the simple sequence in table 4.

Table 4. Sample Sequence

No	Example of command sequences
1	PROCEDURE Initialize_CMG IS
2	ON_ERROR PERFORM CMG_Recovery_Sequence
3	Select_Mode Station TO Normal
4	Set_upper_temp_limit CMG_Heater1 TO 120
5	WAIT 35 SECONDS
6	VERIFY Status OF CMG_Heater1 EQUALS ON WITHIN 10 SECONDS
7	THEN
8	Power_on CMG1
9	PERFORM CMG1_Test_Sequence
10	OTHERWISE
11	Issue CMG_Heater_Failure Message
12	END VERIFY
13	RETURN
14	END Initialize_CMG

Table 5. Example Command Sequencer -- Initialize_CMG

Seq. No.	Command <Mandatory> [Optional]	Delta Time	Time SEC.	Text Description	Seq. Step
1	ATTRIBUTE_WRITE <station> <select_Mode> <Normal>	yes	0	Selected_Mode of Station set to Normal. Initializing CMG	1,3
2	initialize<CMG> <ON_ERROR_Sequence>	yes	0	Scheduling the CMG_Recovery_Sequence when ever there is a CMG error	2
3	ATTRIBUTE_WRITE <CMG> <Heater1> <upper_Temp_Limit> [120]	yes	1	Set the upper limit on CMG Waiting 35 seconds	4
4	initialize <CMG> <VERIFY>	yes	36	Starting the verification of the status of CMG Heater 1 is heating within 10 seconds.	6
5	terminate <CMG> <VERIFY>	yes	46	Finished the verification period on the status of CMG Heater 1	6
6	initialize <CMG> <Heater_failure Message>	yes	47	Enabling the Heater 1 failure message if CMG Heater1 is not heating	11

Table 6. Example Command Sequencer -- Power_and_Test_CMG

Seq. No.	Command <Mandatory> [Optional]	Delta Time	Time SEC.	Text Description
1	ACTION_WRITE <CMG1> <RPC#1> <ON>	yes	0	Power applied to CMG1
2	initialize <CMG1> <Test_Sequence>	yes	1	Initialized the CMG1 test sequence.
3	terminate <Initialize_CMG>	yes	2	Finished CMG initialization

Table 7. Example Discriminator -- ON_ERROR

Assertion Parameters	Discriminator Construct	MADPU <address>
CMG1_Error, Heater1_Error, CMG1_Power_Error	(CMG1_ERROR=NOT{ }) OR (Heater1_Error=NOT{ }) OR (CMG1_Power_Error=NOT{ })	initialize <CMG> <Recovery_Sequencer> <CMG1>

Table 8. Example Discriminator -- Verify_Heater 1 ON

Assertion Parameters	Discriminator Construct	MADPU <address>
Heater1_OperationalState,	(Heater1_OperationalState =enabled)	initialize <Power_and_Test CMG> [CurrentStep] [1]

Table 9. Example Discriminator -- Heater_Failure

Assertion Parameters	Discriminator Construct	MADPU <address>
Heater 1_OperationalState	(Heater1_OperationalState = disabled)	ACTION_WRITE <Crew><heater1_Failure > {disabled} ACTION_WRITE <Ground> <heater1_Failure> [disabled]

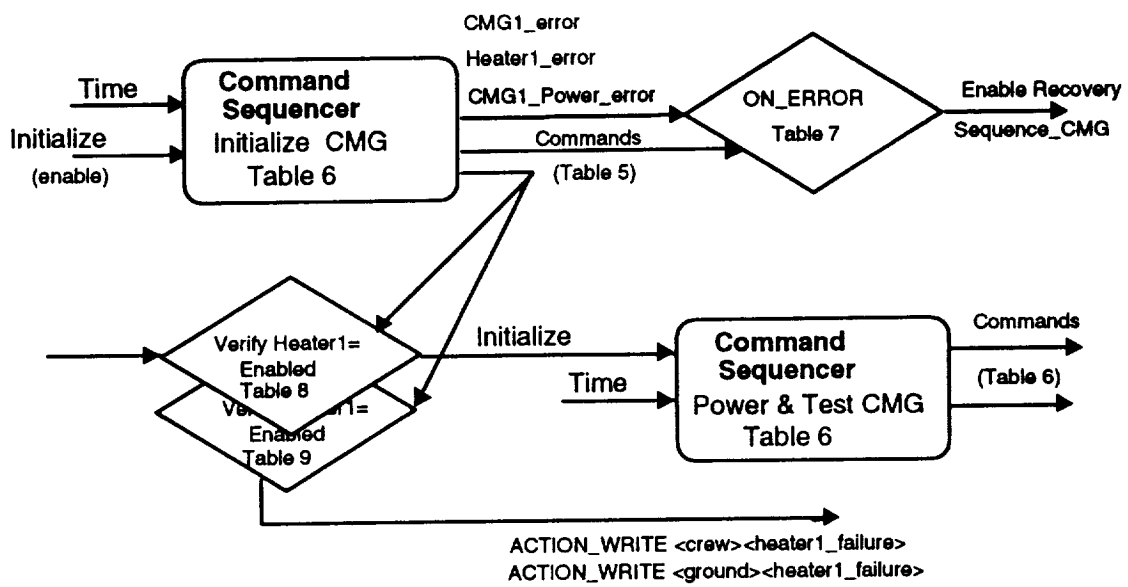


Figure 16. Sample Command Sequence

E.2 The Space Station Event and Message Discriminator Object

E.2.1. Requirements for the Discriminator

The discriminator shall provide the following characteristics:

- The ability to accept events, attribute value assertions, or notifications and discriminate them depending upon the discriminator construct (i.e., a logic statement on attribute values, events, and notifications with capability to group the logical statements with "ORs" and "ANDs")
- The ability to forward or output an event or notification to a specific destination
- The ability to send a specified management application protocol data unit (MAPDU) to the specified destination
- The ability to have an authorized management system (i.e., crew, SSCC, POIC) change the attributes of the discriminator by changing attribute values. (For example, changing the discriminator construct.)
- The ability for the authorized management system to examine the usage, operation, availability, and administrative states of the discriminator (These states are defined in ISO/ICE IS 10164 parts 1 and 2.)
- The ability to schedule the discriminator
- The ability to initiate, terminate, suspend, and resume the discriminator
- The ability to have a user friendly interface when naming the discriminator or when commanding changes to a set of discriminators

E.2.2. The Discriminator Model

The discriminator is a management support object that allows management operations of events and notifications relating to other managed objects.

The discriminator has two logical parts. The first logical part is the discriminator input object. The second logical part is the discriminator output object. The discriminator input object is a conceptual object whose function is to receive or to collect information from the

available data base and to perform the discrimination function. The second logic part of the discriminator sends specified information to another object instance. The attributes of the conceptual output object are the parameters of a management application protocol data unit (MAPDU) or a potential MAPDU that may be sent from the discriminator output object. Discrimination is only performed upon attribute value assertions, received notifications or events that have defined matching rules. The discriminator output object forwards a MAPDU or a potential MAPDU to a destination that is either remote or internal to the local managed system.

- The ability to accept events, attribute value assertions, or notifications and discriminate them depending upon the discriminator construct (i.e., a logic statement on attribute values, events, and notifications with capability to group the logical statements with "ORs" and "ANDs")
- The ability for the discriminator constructs to be based upon equals, less than greater than, not equal to, in range, in set, out of range, and not the events, attribute value assertions, or notifications
- The ability to forward or output an event or notification to a specific destination
- The ability to send a MAPDU to the specified destination
- The ability to have an external authorized management system (i.e., crew, SSCC, POIC) change the attributes of the discriminator by changing attribute values (For example changing the discriminator construct.)
- The ability for the external authorized management system to examine the usage, operation, availability, and administrative states of the discriminator (These states are defined in ISO/ICE IS 10164 parts 1 and 2.)
- The ability to schedule the discriminator
- The ability to initiate, terminate, suspend, and resume the discriminator
- The ability to determine the status of all the instances of discriminators through telemetry object lists

E.2.2.1 The Normal Operation of the Discriminator

The figure 17 illustrates the discriminator.

The discriminator has attributes that characterize its behaviour. The attributes of the discriminator are as follows:

- The **discriminatorID** attribute identifies the instance of the discriminator. The discriminatorID is of type File_ID
- The **discriminatorConstruct** attribute specifies the logical test on the information that is processed by the discriminator construct.
- The **administrativeState** attribute represents the *locked* and *unlocked* states. When the discriminator is *locked*, the logical test can be replaced.
- The **operationalState** attribute represents the object *enabled* and object *disabled* states of the discriminator. *Enabled* means the discriminator has been initialized and ready for use. *Disabled* means the discriminator is inoperable.
- The **usageState** attribute represents the *idle* and *busy* states. The *idle* state means the discriminator is suspended. The *busy* state means the discriminator is object enabled and being fully used.
- The **availabilityStatus** attribute represents the schedule of the discriminator. The availability status attribute has the values of *off-duty* or *on-duty*. *Off-duty* indicates that the discriminator is currently not scheduled. The availability status attribute is only used if the discriminator is instantiated with external or internal scheduling capabilities.
- The external scheduling of a discriminator is represented by a **schedulerName** attribute. When the attribute is not NULL, then the state of the scheduler determines the availability of the discriminator construct. The schedulerName is a relationship attribute (a pointer) to a sequencer that controls when the discrimination construct is to be performed. Sequencer managed objects are defined in section E.3.
- The **outputMapdu** attribute specifies the MAPDU to be sent by the discriminator construct.

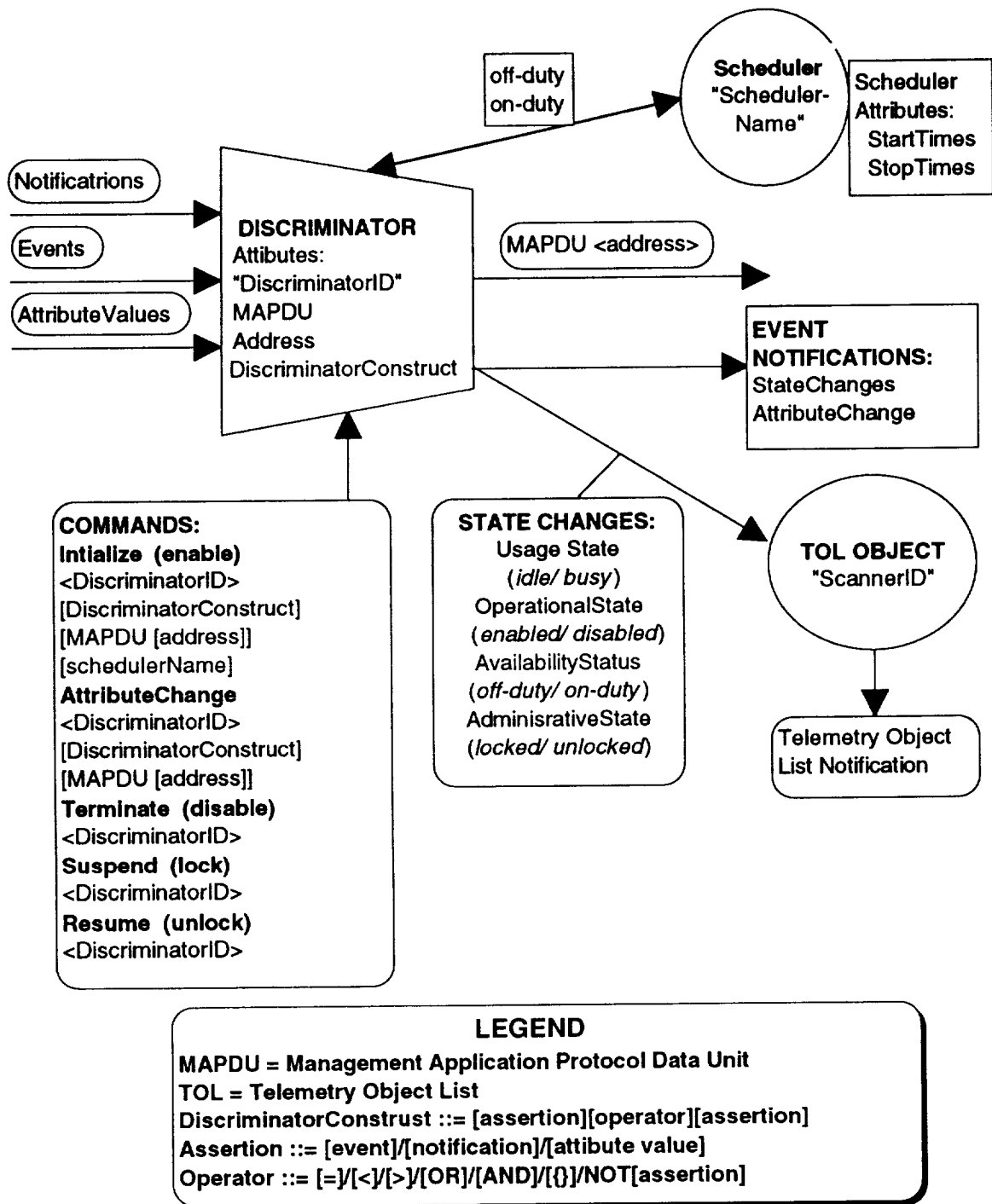


Figure 17. The Discriminator Model

E.2.2.2 The Discriminator Behaviour

The discriminator construct is a filtering mechanism which acts on attributes of the discriminator inputs. The discriminator construct is a set of one or more assertions about the presence or values of observed attributes. If the discriminator construct involves more than one assertion, the assertions are grouped together using logic operations.

The discriminator construct can specify a test for equality and inequality conditions of attributes, a test for the presence of attributes, and the negation of any of the conditions. Multiple conditions may be combined by means of "AND" or "OR" operators. When an attribute for which an attribute value assertion is present in the discriminator construct, is absent in a discriminator input object to be tested, the results of the test on that attribute value assertion shall be evaluated FALSE.

Given that specified conditions (such as an external schedule or sequencer) are satisfied, a discriminator that contains an empty discriminator construct will evaluate to TRUE for any set of discriminator input object attributes.

For the discriminator, if the discriminator construct evaluates to TRUE, and the discriminator is in the *unlocked* and *enabled* state, and the availability status is not *off-duty*, then the discriminator emits the output MAPDU specified by the **outputMapdu** or if the MAPDU is NULL, then the discriminator sends the input MAPDU to the specified destination.

If the discriminator is in the *locked* state, then the discriminator accepts attribute value changes to the discriminator attributes.

If the discriminator is in the *locked* or in the *off-duty* availability status, then the discriminator input object will not be processed by that discriminator.

E.2.2.3 The Discriminator Actions

The discriminator shall have service actions to do the following:

- Terminate the discriminator: The termination takes the form of a DMS ACTION WRITE command (i.e., a Common Management Information Service [CMIS] delete) and results in the discriminator shutting down. During shutdown the discriminator processes the current inputs and finishes the current input discrimination and outputs the event or forwarded message in the form of a

MAPDU. The discriminator completes its current behaviour and transitions to the disabled state.

- Initialize the discriminator: The initialization action takes the form of DMS ACTION WRITE command (i.e., a CMIS create) and has parameters⁴⁷ to specify all the discriminator attributes.
- Suspend the discriminator: The suspend action is mapped to a DMS ACTION WRITE command (i.e., a CMIS SET) which writes the *locked* administrative state. When the discriminator is locked its behaviour is suspended. In the suspended discriminator the attributes of the discriminator can be changed.
- Resume the discriminator: The resume action is mapped to a DMS ACTION WRITE command (i.e., a CMIS SET) to write the *unlocked* administrative state. When the discriminator transitions to the *unlocked* state, the discriminator resumes its behaviour.
- Change the discriminator: The attribute change action is mapped to the DMS ACTION WRITE command (i.e., a CMIS SET) to replace the attributes of the discriminator construct, the discriminator output MAPDU, or discriminator output destination address. The parameters of the DMS WRITE command map to the attributes of the discriminator.

E.2.3. The Discriminator GDMO

The discriminator object class is used to define the criteria for controlling management service. The semantics of the object class, namely its attributes and behaviour are described in the behaviour description. The following description follows the requirements of ISO international standard ISO/IEC 10165 part 4 -- Generic Definition of Managed Objects (GDMO).

discriminator MANAGED OBJECT CLASS

DERIVED FROM ---the space station's top object class:top;	
CHARACTERIZED BY	discriminatorPackage;
discriminatorPackage	PACKAGE
BEHAVIOUR DEFINITIONS	discriminatorClassBehaviour;
ATTRIBUTES	
administrativeState	GET,---ATTRIBUTE_READ

⁴⁷ Parameters of an action are the bit fields of the command.

availabilityStatus	PERMITTED VALUES - Attribute-
ASN.1 Module.DiscriminatorAvailability	GET, --- Read
destination Address	GET-REPLACE, --- read or write
discriminatorConstruct	GET-REPLACE,
discriminatorID	GET,
operationalState	GET,
usageState	GET,
outputMapdu	GET-REPLACE;

ACTIONS

Initialize	-	-- MAPS TO DMS ACTION_WRITE with
discriminatorIDParameter	-- Mandatory (M),	
discriminatorConstructParameter	-- Optional (O),	
outputMapduParameter	-- O,	
AttributeChange	---	MAPS TO DMS ACTION_WRITE with
discriminatorIDParameter	-- M,	
discriminatorConstructParameter	-- O,	
outputMapduParameter	-- O,	
Terminate	---	MAPS TO DMS ACTION_WRITE with
discriminatorIDParameter	-- M,	
Suspend	--	MAPS TO DMS ACTION_WRITE with
discriminatorIDParameter	-- M,	
Resume	---	MAPS TO DMS ACTION_WRITE with
discriminatorIDParameter	-- M;	

NOTIFICATIONS

stateChange,-- See State Change notification table of ISO/IEC 10164-2.
 -- This notification maps to a DMS event notification.

attributeValueChange;;;-- See table 5 Attribute Value Change
 Reporting parameters in ISO/IEC 10164-1. -- This notification
 maps to a DMS event notification.

-- The above event notifications are defined in {recommendation X.731 [4] | ISO/IEC 10164-2], State Management Function and i{Recommendation X.731 [3] | ISO/IEC 10164-1], Object Management Function. These event notifications map to DMS standard service events.

-- The issuing of the above event notifications could be controlled by the application of a discriminator acting as an event notification forwarding discriminator. An alternate but expensive use of bandwidth method of reporting the state changes would be to use a TOL notification. The administrative, availability, usage, and operational states are observable attributes in the RODB. A DMS standard service could scan

with ATTRIBUTE SYNTAX Attribute-ASN.1 Module.DestinationAddress;
MATCHES FOR equality;
REGISTERED AS (smi2AttributeID 89); -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Discriminator Construct

The semantics of the discriminatorConstruct attribute type are specified in the Discriminator Model.

discriminatorConstruct ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1 Module.DiscriminatorConstruct;-
REGISTERED AS (smi2AttributeID 90); -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

discriminatorID

The semantics of the discriminator ID attribute type is used in naming instances of discriminator object class. The discriminatorID is to be of the type, FILE_NAME.

discriminatorID ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-ASN1Module.DiscriminatorID;
MATCHES FOR Equality, Substrings;
REGISTERED AS {smi2AttributeID 1}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Usage State

The usageState attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 39}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Operational State

The operationalState attribute is specified in Recommendation X.731 ISO/IEC 10164-2,
REGISTERED AS {smi2AttributeID 35}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Availability Status

The availabilityStatus attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 33}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Administrative State

The semantics of the administrativeState attribute type are specified in the Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 31}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

The output MAPDU

The semantics of the outputMapdu is described in the Discriminator Model.

outputMapdu ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.DiscriminatorSyntax;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Scheduler Name

The semantics of the SchedulerName attribute type are specified in the Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 102}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

DMSModule.DiscriminatorSyntax
outputMapdu::= CHOICE OF
 DMS ACTION_WRITE with parameters
 DMS ATTRIBUTE_WRITE with parameters
 --- DEFAULTS to NULL;

E.2.4 The DMS IRD Template

This section provides the DMS template per the DMS IRD. The template includes forms for the DMS data objects and the managed object actions.

DISCRIMINATOR DATA OBJECTS

Attributes

discriminatorID

Description:

The **discriminatorID** attribute identifies the instance of the discriminator

Constrains: The discriminatorID will meet the object naming requirements and valid name.

RODB Access: Read/Write

RODB Processing: None

MODB Processing: None

discriminatorConstruct

Description:

The **discriminatorConstruct** attribute specifies the logical test on the information that is processed by the discriminator

Constrains: Processing per Event or Attribute constrains, semantics and processing per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read/Write

RODB Processing: None

MODB Processing: None

administrativeState

Description:

The **administrativeState** attribute represents the *locked* and *unlocked* states. When the discriminator is *locked* the discriminator construct attribute can be replaced.

Constrains: Semantics ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read

RODB Processing: None

MODB Processing: None

operationalState

Description:

The **operationalState** attribute represents the object *enabled* and object *disabled* states of the discriminator. *Enabled* means the discriminator has been initialized and ready for use. *Disabled* means the discriminator is inoperable.

Constrains: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read
RODB Processing: None
MODB Processing: None

usageState

Description:

The **usageState** attribute represents the *idle* and *busy* states. The *idle* state means the discriminator is suspended. The *busy* state means the discriminate is object enabled and being fully used.

Constrains: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read
RODB Processing: None
MODB Processing: None

availabilityStatus

Description:

The **availabilityStatus** attribute represents the schedule of the discriminator. The availability status attribute has the values of *off-duty* or *on-duty*. *Off-duty* indicates that the discriminator is currently not scheduled. The availability status attribute is only used if the discriminate is instantiated with external or internal scheduling capabilities.

Constrains: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read
RODB Processing: None
MODB Processing: None

schedulerName

Description:

The external scheduling is represented by a **schedulerName** attribute. When the attribute is not NULL, then the state of the scheduler determines the availability of the discriminator.

Constrains: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read/Write
RODB Processing: None
MODB Processing: None

outputMapdu

Description:

The **outputMapdu** attribute specifies the MAPDU to be sent by the discriminator.

Constrains: Per DMS ACTION and DMS WRITE specifications

RODB Access: Read/Write

RODB Processing: None

MODB Processing: None

Actions:

Initialize --- MAPS TO DMS ACTION_WRITE with
<discriminatorIDParameter>,
[discriminatorConstructParameter],
[outputMapduParameter],
AttributeChange --- MAPS TO DMS ACTION_WRITE with
<discriminatorIDParameter>,
[discriminatorConstructParameter],
[outputMapduParameter],
Terminate --- MAPS TO DMS ACTION_WRITE with
<discriminatorIDParameter>,
Suspend --- MAPS TO DMS ACTION_WRITE with
<discriminatorIDParameter>,
Resume --- MAPS TO DMS ACTION_WRITE with
<discriminatorIDParameter>;

E.3 The Space Station Command Sequencer Object

E.3.1 Requirements for the Command Sequencer

The Command Sequencer shall provide the following characteristics:

- The ability to replace a single step within a command sequence supplied by action writes to the sequencer (Note the replacement of NULL steps allows the capability to supplement behaviour.)
- The ability to replace a command sequence as a parameter supplied by an attribute write command (Command sequences are executed in step order or to a specified time-line.)
- The ability to store absolute times or delta times from the first step of a sequence
- The ability to determine status of the sequences to determine operational state, enabled (initiated), or disabled (terminated) and determine the command sequence step, usage state, idle (suspended), busy (executing)
- The ability to initiate, terminate, suspend, resume services
- The ability to suspend for a period of time or at a given sequenced command
- The ability to commence the sequence at any single step
- The ability to determine status of all of the instances of command sequences through telemetry object list notifications
- The ability to send notification of sequencer state changes and attribute changes when they occur to any authorized destination.
- The ability to have a user friendly interface when naming the command sequences or when commanding the command sequences.

E.3.2 The Command Sequencer Model

The Command Sequencer is a management support object that provides management and control of sequences of commands called procedures. One or more of these procedures can

be stored in files that are linked to the Command Sequencer. The sequencer executes the steps of the procedure either in the supplied order or in accordance with a specified time-line. A special ability of the Command Sequencer is that it can be sent one or more attribute write commands to replace a procedure in orbit-time⁴⁸.

E.3.2.1. The Normal Operation of the Command Sequencer

Figure18 illustrates the Command Sequencer. The Command Sequencer needs the following attributes to characterize its behaviour:

- The **operationalState** attribute to indicate the object *enabled* (initialized) and the object *disabled* (terminated) states of the sequencer.
- The **usageState** attribute to indicate the functioning of the Command Sequencer. The usage state includes *idle* (to indicate suspended Command Sequencer activity) and *busy* (to indicate execution activity).
- The **schedulerName** relationship attribute to identify an external scheduler (i.e., another sequencer) that can control the times when the sequencer is available and *on-duty* or *off-duty*. The sequencer is *on-duty* as scheduled by an external scheduler or by an internal schedule condition. *Off-duty* indicates that the sequencer has been stopped by the schedule.
- The **availabilityStatus** attribute to indicate when the sequencer is *on-duty* as scheduled by an external scheduler or internal scheduler. The attribute values of this attribute are *on-duty* and *off-duty*.
- The **currentStep** attribute to indicate the current command to be or being executed.
- The **fileList** attribute to indicate the command files the Command Sequencer was preloaded with when it was initialized.
- The **holdStep** attribute to indicate the command sequence number at which the sequencer usage state will become *idle*.
- The **startTime** attribute indicates when the command sequence became enabled.

⁴⁸ Orbit-time is the period of time associated with a few orbits. This term is used in preference to *real-time* because the context of real-time implies a fast reaction time.

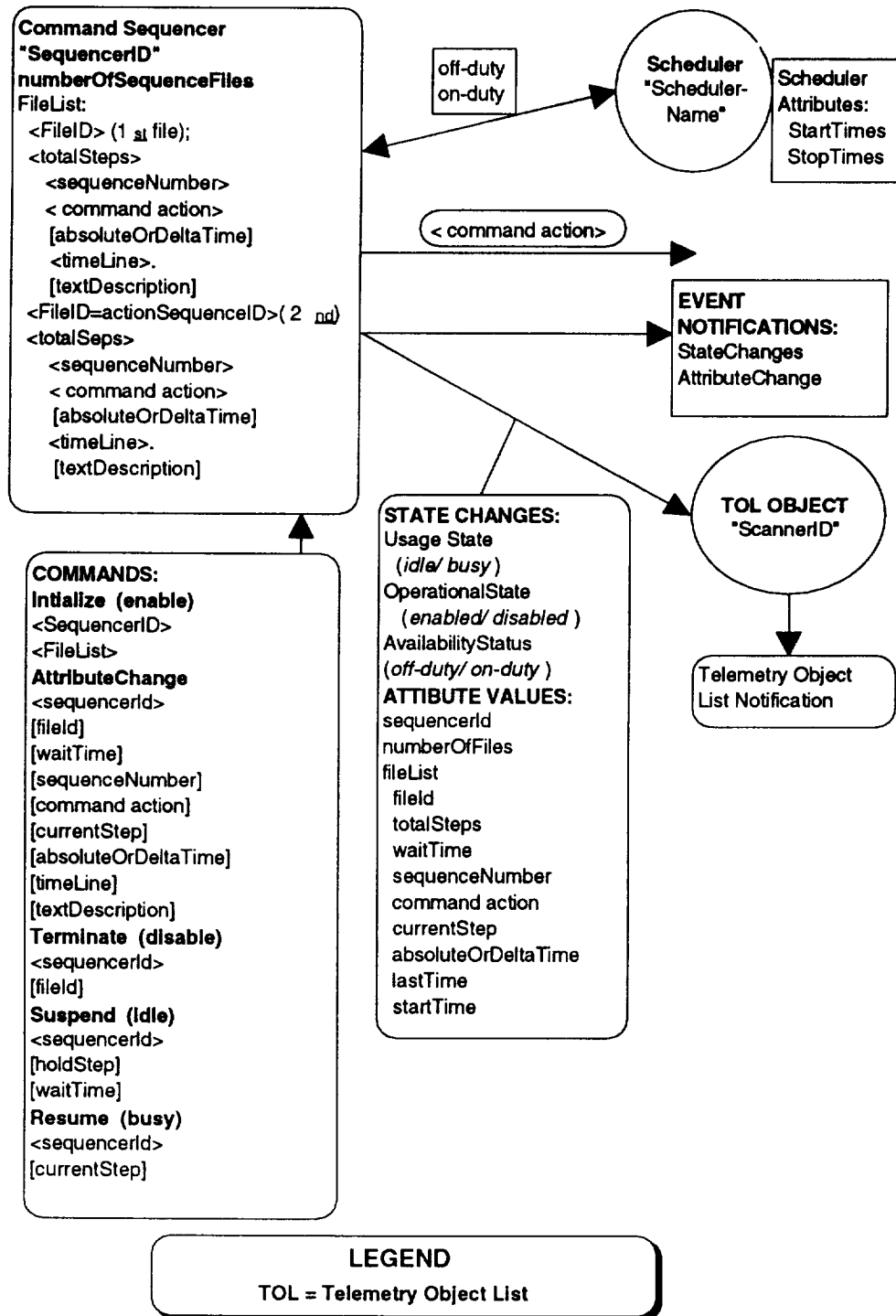


Figure 18. The Command Sequencer Model

- The **lastCommandTime** attribute indicates the time the last command in this command sequence was executed.
- The **numberOfSequenceFiles** attribute indicates the number of stored files in this instance of the Command Sequencer.
- The **waitTime** attribute indicates the time the sequencer will *idle* before executing. If not off-duty, the sequencer at the completion of the wait time will transition from *idle* (suspended) to *busy* (executing).
- The **sequencerID** to identify the instance of the sequencer.
- A set of the **sequencerID** attribute values are the **actionSequencerIDs** assigned to indicate that the commands were loaded by **loadSequence** commands received by the Command Sequencer.
- The **fileList** attribute has attributes to identify the parameters of the records of the files. These attributes of the command record include:
 - The **fileID** attribute identifies the file instances specifying the activity.
 - The **totalSteps** attribute identifies the size of the sequence in the file.
 - The command **sequenceNumber** attribute identifies the order of the steps in the file.
 - The **commandAction** attribute provides the parameters of an action write command or an attribute read or write command. One **commandAction** value (a command with all its parameter values) is associated with each sequence number.
 - The **absoluteOrDeltaTime** attribute indicates if the time-line attribute value is an absolute time given in GMT, or a delta time in integer seconds from the first command in the list. One **absoluteOrDeltaTime** attribute value is associated with each sequence number.
 - The **timeLine** attribute identifies the time the commands are to be issued by the Command Sequencer. One **timeLine** attribute value is associated with each sequence number.

- The **textDescription** attribute provides the text descriptions of the individual commands. One **textDescription** attribute value (a string) is associated with each sequence number.

E.3.2.2. The Command Sequencer Waiting Services

The Command Sequencer has two waiting services which are established through service commands. These service commands are initialization and holding. The initialization command is started with an **initialize** sequencer command. The initialize sequencer command specifies the **fileList** of the Command Sequencer (Note one or more of the **fileList** identifiers can be an **actionSequenceID**). If the supplied file list identification attribute is NULL, then the sequencer behaviour defaults to that of an **actionSequenceID** sequencer whose commands are loaded as the result of receiving **attributeChange** sequencer command actions. The sequencer becomes object *enabled* after receiving an **initialize** action that has a **currentStep** parameter. If the sequencer is not *off-duty*, then the sequencer starts and becomes *busy* at the specified **currentStep** sequence number. The execution is stopped with a **terminate** action.

The holding service is started with a **suspend** action command with an optional **waitTime** parameter or an optional **holdStep** parameter. The holding service is released by a **resume** action or the countdown of **waitTime**.

E.3.2.3. The Command Sequencer Behaviour

The command sequence behaviour is characterized by the following descriptions:

Upon receiving an **initialize** action command that includes a **fileList** parameter, a **currentStep** parameter, an optional **schedulerName** parameter, and an optional **waitTime** parameter for each file in the file list, then the Command Sequencer is instantiated with the content of the files and the values of the parameters. The files contain command sequence numbers, commands, time-lines, delta or absolute time indicators, and the text descriptions of the commands. These parameters of the files form the command record object that is used for command logging. After initialization, the DMS standard services are established for the instantiated Command Sequencer. The required DMS services include the establishment of a checkpoint journal, telemetry object list (TOL) scanner, a TOL notification, and a command log. These services are commanded by sending initialization actions or **attributeChange** actions (DMS ACTION WRITE) to the DMS System Manager. During the initialization, the sequencer usage state is *idle*, the operational state is object *disabled*, and the availability state (*off-duty* or *on-duty*) is determined by the operational state of the optional scheduler. After

initialization the sequencer usage state becomes *enabled* and the operational state becomes either *idle* or *busy* as determined by the **waitTime** or the **availabilityStatus** for each file.

Upon receiving a **resume** action with an optional **currentStep** parameter, or an optional **waitTime** parameter, the Command Sequencer starts at the specified sequence number after the specified **waitTime** and executes the commands in time-line order as scheduled from the first command in the file of command sequences. If the availability status is *on-duty*, then the usage state becomes *busy* after the **waitTime**. During the wait time the usage state is *idle* and the operational state is *enabled*. If the **resume** command does not contain a **currentStep** parameter the Command Sequencer starts at the last executed step or if it has never been *busy* the **currentStep** defaults to the first step.

The *enabled*, *busy* Command Sequencer samples the current time and stores the value in the **startTime** attribute. The **startTime** attribute value is used as the dependent time-off-set for the remaining commands if they have delta time values. If the commands have absolute times identified, then the sequencer uses current time to countdown to the time when the commands are issued. If any stored command as an absolute time less than the current time, then that command is skipped and not executed.

The sequence continues in the *busy* usage state until it is commanded by a **suspend** action command.

If commanded with a **suspend** action command with a **waitTime** parameter, then the sequencer waits the specified time and continues. Delta time commands are suspended by the **waitTime**, and upon resuming the *busy* usage state the sequencer stores a new **startTime** value to be used in determining the time-line. Absolute-times commands passed over during the **waitTime** are not issued by the sequencer. If commanded with a **suspend** action command with a **holdStep** parameter, the sequencer enters the *idle* usage state after executing the command and remains *idle* until it receives **resume** action command.

If a **resume** command with a **currentStep** parameter is received without a **waitTime** parameter, then the sequencer goes to the new step on the next command and continues from that point. The **startTime** attribute is updated when new **currentStep** is issued.

If an **attributeChange** action command is received, then the sequencer using the **attributeChange** action command parameters replaces or adds the specified stored command. If the **attributeChange** action command replaces a NULL command into the sequencer, then the **totalSteps** attribute for the sequence is updated.

The sequencer remains in the operational or usage state that it was in at the time of the **attributeChange** action command was received. The reception of the **attributeChange** action command does not affect the ongoing current behaviour of the sequencer.

The sequencer continues until after the last command is executed or if a terminate action command is received. Then the usage state transitions to *idle* and the operational state transitions to *disabled* (i.e, terminated = *idle* AND *disabled*).

The sequencer always completes its current started behaviour. It can be changed only during the periods between issuing commands.

E.3.2.4. The Command Sequencer Actions

The Command Sequencer has the following actions which command its behaviour:

- The **attributeChange** action command has parameters indicating the **fileID**, the command **sequenceNumber**, the command or its identification (If a command tag or command identification is supplied, then DMS is expected to supply the command to be included with all parameters set to the default values.), time-line (either absolute or delta times from the first command), and text descriptions.
- The **resume** action command has parameters to specify the sequencer ID, an optional **waitTime**, and an optional **currentStep**.

E.3.2.5. The Command Sequencer Notifications and TOL

The Command Sequencer optional generates notifications for state and attribute changes. The Command Sequencer can generate these notifications by using the event notification services of DMS. A state change event sends the state change notification, and an attribute change event sends the attribute change notification. Also, the Command Sequencer can be supported by DMS standard services that generate a TOL to report the state changes and the attribute changes. The sequencer TOL would include the following parameters:

- The total number of sequencer files scanned in the TOL.
- The list of **fileIDs** from which the sequencers were instantiated.
- The **totalSteps** in each file.
- The list of **usageStates** (*idle/busy*) for each **fileID** instance.
- The list of **operationalStates** (*enabled/disabled*) for each **fileID** instance.
- The list of **availabilityStates** (*on-duty/off-duty/NULL*) for each **fileID** instance.
- The list of schedulerNames/NULL for each **sequencerID** instance.

- The list of **startTimes** for each **fileID** instance.
- The list of requested **holdStep** numbers for each **fileID** instance.
- The list of **waitTimes** for each **fileID** instance.
- The list of the **currentSteps** of each **fileID** instance.
- The list of the **currentTimes** of each executing command of each **fileID** instance.

E.3.3 The Command Sequencer GDMO

The Command Sequencer object class is used to define the commanding of management operations. The semantics of the object class, namely its attributes and behaviour are described in the behaviour description (section 2.). The following description follows the templates of ISO international standard ISO/IEC 10165 part 4 -- Generic Definition of Managed Objects (GDMO).

The Command Sequencer

Command Sequencer MANAGED OBJECT CLASS

```
DERIVED FROM ---the space station's top object class:top;
CHARACTERIZED BY                                sequencerPackage
sequencer      PACKAGE
BEHAVIOUR DEFINITIONS                          sequencerClassBehaviour;
ATTRIBUTES
    availabilityStatus                          PERMITTED VALUES - Attribute-
        ASN.1 Module.DiscriminatorAvailability GET,
    sequencerID                                GET,
    operationalState                            GET,
    usageState                                  GET,
    fileList                                   GET-REPLACE,
    currentStep                                GET-REPLACE,
    holdStep                                   GET-REPLACE,
    waitTime                                   GET-REPLACE,
    startTime                                  GET,
    lastCommandTime                            GET,
    numberOfSequenceFiles                      GET-REPLACE,
ACTIONS
    initialize      --- MAPS TO DMS ACTION _WRITE with
        sequencerID      ---Mandatory (M)
        fileList,        ---optional (O)
        waitTime,        --- O
        SchedulerName    --- O
    attributeChange  --- MAPS TO DMS ATTRIBUTE_WRITE with
        sequencerID,      ---Mandatory (M)
        fileID,           ---Mandatory (M)
        sequenceNumber,   --- M
        command,          --- M
        absoluteOrDeltaTimeInd, --- M
        timeLine,         --- M
```


textDescription,	---	O
currentStep,	---	O
resume	---MAPS TO DMS ACTION_WRITE with	
sequencerID,	---	Mandatory (M)
fileID,	---	Mandatory (M)
currentStep,	---	O
waitTime,	---	O
terminate	--- MAPS TO DMS ACTION_WRITE with	
sequencerID,	---	Mandatory (M)
fileID,	---	Mandatory (M)
suspend	--- MAPS TO DMS ACTION_WRITE with	
sequencerID,	---	Mandatory (M)
fileID,	---	Mandatory (M)
waitTime,	---	O
holdStep;	---	O

NOTIFICATIONS

stateChange,-- See State Change notification table of ISO/IEC 10164-2.
This notification maps to DMS standard event services.

attributeValueChange;;;--- See table 5 Attribute Value Change
Reporting parameters in ISO/IEC 10164-1. This notification
maps to DMS standard event services.

-- The above event notifications are defined in {recommendation X.731 [4] | ISO/IEC 10164-2}, State Management Function and in {Recommendation X.731 [3] | ISO/IEC 10164-1}, Object Management Function.

-- The issuance of the above event notifications could be controlled by the application of a discriminator acting as an event notification forwarding discriminator. An alternate but expensive use of bandwidth would report the state changes in a TOL notification. The administrative, availability, usage, and operational states are observable attributes in the RODB. A DMS standard service could scan these attributes and output a Telemetry Object List (TOL) for sequencers. The Telemetry Object List (TOL) notification reports parameters of each sequencer and each file in the sequencers. The TOL includes the total number of sequencer files, and the **sequencerIDs**. For each **sequencerID** the TOL includes the **fileIDs**. The TOL includes for each **fileID** the **totalSteps**, **startTime**, **holdStep** sequence number, the **waitTime**, **currentStep**, and **currentTime**.

CONDITIONAL PACKAGES

schedulerName PRESENT IF the object supports an external scheduler;
REGISTERED AS {DSSRDB.Object_ID: ??};

Package Definitions

External Scheduler

The semantics of the external scheduler package are described in Recommendation X.735 [7] [ISO/IEC 10164-5] and Recommendation X.735 [8] [ISO/IEC 10165-6]-- The scheduler package attributes map into DMS ObjectIDs.

```
schedulerName          PACKAGE
  ATTRIBUTES
    schedulerName  GET-REPLACE;
REGISTERED AS {smi2Package 2}
```

Behaviour definitions

```
sequencerBehaviour  BEHAVIOUR          DEFINED AS
  ---The behaviour of the sequencer is described by the command sequence model (section
  2.2) in this document. -- The behaviour is mapped to the application module that provides
  the procedure of the sequencer.
```

Attributes definitions

Usage State

The usageState attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 39}; -- The usage state attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Operational State

The operationalState attribute is specified in Recommendation X.731 ISO/IEC 10164-2,
REGISTERED AS {smi2AttributeID 35}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Availability Status

The availabilityStatus attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 33}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Scheduler Name

The semantics of the SchedulerName attribute type are specified in the Recommendation X.731 ISO/IEC 10164-2.

REGISTERED AS {smi2AttributeID 102}; -- The attribute is mapped to a DSSRDB.Object_ID_T.

Current Step

The semantics of the currentStep is described in the Command Sequencer model.

currentStep ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Hold Step

holdStep ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??}

File List

fileList ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Last CommandTime

lastCommandTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Number Of Sequence Files

numberOfSequenceFiles ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Start Time

startTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Sequencer ID

sequencerID ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality, Substrings;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Wait Time

waitTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.CommandSequencer;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

DMSModule.CommandSequencer

AbsoluteTime::= GMT
currentStep::= INTEGER
HoldStep::= INTEGER
FileList::= SEQUENCE_OF Files
Files::= SEQUENCE_OF
fileID FileID
totalSteps TotalSteps
commandSteps
absoluteOrDeltaTimeInds
timeLines TimeLine

DMS ACTION
TimeFlag

PRINTABLE TEXT

textDescriptions

FileID::= File_Name

DMS ACTION::= CHOICE OF {DSSRDB.ATTRIBUTE_WRITE,
DSSRDB.ACTION_WRITE}

DeltaTime::= INTEGER --- represent the integer number of seconds to the next time

LastCommandTime::= GMT

NumberOfSequenceFiles::= INTEGER

StartTime::= GMT

SequenceID::= DSSRDB.ATTRIBUTE_T

TimeFlag::= BINARY ---- one indicate absolute times, zero indicates delta times

TimeLine::= CHOICE OF
AbsoluteTime
DeltaTime

TotalSteps::= INTEGER

WaitTime::= INTEGER --- represent the integer number of seconds to wait

E.3.4 The DMS IRD Template

This section provides the DMS template per the DMS IRD. The template includes DMS forms for the DMS data objects and DMS ACTION WRITES.

Attributes

sequencerID

Description:

The **sequencerID** attribute identifies the instance of the sequencer

Constraints: The sequencerID will meet the object naming requirements and it shall be a valid name. The name shall be of the type File_Name.

RODB Access: Read/Write

RODB Processing: None

MODB Processing: None

operationalState

Description:

The **operationalState** attribute represents the object *enabled* and object *disabled* states of the discriminator. *Enabled* means the sequencer has been initialized and commanded to issue commands. *Disabled* means the WaitTime is inoperable.

Constraints: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read

RODB Processing: None

MODB Processing: None

usageState

Description:

The **usageState** attribute represents the *idle* and *busy* states. The *idle* state means the discriminator is suspended. The *busy* state means the object is enabled and being fully used.

Constrains: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read

RODB Processing: None

MODB Processing: None

availabilityStatus

Description:

The **availabilityStatus** attribute represents the schedule of the sequencer. The availability status attribute for the values of *off-duty* or *on-duty*. *Off-duty* indicates that the discriminator is currently not scheduled. The availability status attribute is only used if the sequencer is instantiated with external or internal scheduling capabilities.

Constrains: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read

RODB Processing: None

MODB Processing: None

schedulerName

Description:

The external scheduling is represented by a **schedulerName** attribute. When the attribute is not NULL, then the state of the scheduler determines the availability of the Command Sequencer. The schedulerName shall be of type File_Name.

Constrains: Semantics per ISO/IEC 10165-2 and ISO/IEC 10164

RODB Access: Read/Write

RODB Processing: None

MODB Processing: None

startTime

Description:

The **startTime** attribute reports the time when the sequence started.

Constrains: The start time shall be reported as GMT.

RODB Access: Read
RODB Processing: None
MODB Processing: None

lastCommandTime

Description:

The **lastCommandTime** attribute has the value of two when the last command sequence was issued.

Constrains: The **lastCommandTime** shall be reported as GMT

RODB Access: Read
RODB Processing: None
MODB Processing: None

currentStep

Description:

The **currentStep** attribute reports the next command to be issued.

Constrains: INTEGER.

RODB Access: Read/Write
RODB Processing: None
MODB Processing: None

holdStep

Description:

The **holdStep** attribute reports the command step where the command sequence will become idle.

Constrains: INTEGER.

RODB Access: Read/Write
RODB Processing: None
MODB Processing: None

numberOfSequenceFiles

Description:

The **numberOfSequenceFiles** attribute reports the number of fileIDs with command sequences in the sequencer.

Constrains: INTEGER.

RODB Access: Read/Write
RODB Processing: None
MODB Processing: None

fileList

Description:

The **fileList** attribute is sequence of files. Each file is a sequence of records consisting of the following parameters: fileID, totalSteps, CommandSteps, AbsoluteOrDeltaTimeIndicators, a time-line, and a set of text descriptions.

Constraints: Sequence files with of Sequence of command records.

RODB Access: Read/Write

RODB Processing: None

MODB Processing: None

Actions:

```
initialize          --- MAPS TO DMS ACTION_WRITE with
    <sequencerIDParameter>,
    [FileListParameter],
    [waitTimeParameter],
    [SchedulerNameParameter],
attributeChange     --- MAPS TO DMS ACTION_WRITE with
    <sequencerIDParameter>,
    <fileIDParameter>
    <sequenceNumberParameter>
    <commandParameter>
    <absoluteOrDeltaTimeInd,Parameter>
    <timeLineParameter>
    <textDescriptionParameter>
    <currentStepParameter>
resume             --- MAPS TO DMS ACTION_WRITE with
    <sequencerIDParameter>;
    <fileIDParameter>
    [currentStepParameter]
    [waitTimeParameter]
terminate          --- MAPS TO DMS ACTION_WRITE with
    <sequencerIDParameter>
    <fileIDParameter>
Suspend            --- MAPS TO DMS ACTION_WRITE with
    <sequencerIDParameter>
    <fileIDParameter>
    [waitTime]
    [holdStep]
```


APPENDIX F

PROPOSED SPACE STATION OBJECT AND RELATIONSHIP ATTRIBUTES

MITRE recommends the adoption of a set of standard attributes and relationship attributes for use in describing the configurations of the space station systems, elements, and payloads that interface with ISE. This document provides a proposed candidate list for these attributes. The developers of the ISE could assist in this standardization process by using this list to develop a set of standard attributes along with the semantics needed for ISE to perform its system control function.

The following discussion proposes object and relationship attributes for the on-board managed objects⁴⁹ that have attribute values of interest to the ISE. Object attributes are features and characteristics that describe the object or its behaviour. Relationship attributes are object attributes that relate two or more objects, for example, objects operating together in a station mode, back-up objects, or backed-up objects, etc. The attributes described may be included as necessary in the definitions of the managed object classes to describe the state or status of resources important to ISE, the crew, or the ground.

F.1 Object Attributes and Relationship Attributes Requirements

To meet the program requirements (JSC 31000), the ISE needs the capability to perform the system control function for the on-board systems, elements, and payloads. In performing the system control function, it could be determined that derived requirements are the following abilities:

- The ability to relate a remote power circuit breaker (RPC) to a managed object

⁴⁹ Managed objects: A managed object is an abstract representation of the resources of a managed system. The management of these resources requires a management view of the logical and physical identities within the managed system. Managed objects have attributes, actions (commands), notifications, and behaviour, and the managed object behaviour is the result of either changing attribute values, commanding actions to change the managed process, changing the managed object's environment, or changing the behaviour rules associated with the managed object. Examples of managed objects are systems, elements, payloads, orbital replaceable units, standard data processors, mass storage units, pumps, sensors, effectors, etc.

- The ability to relate backed-up and back-up managed objects as necessary in each managed object (This ability may already be available by using the DMS object class, **Logical_Name**.)
- The ability to relate hot-standby and cold-standby to backed-up and back-up managed objects
- The ability to determine if managed objects are members of the station modes (This ability may be not be needed if the mode capabilities of all managed objects are predefined and static.)
- The ability to relate augmented caution and warning services to managed objects within the systems, elements, and payloads
- The ability to relate a set of attribute values to states and statuses of the managed objects (This includes the international standard attributes of availability status, procedural status, and control status.)
- The ability to convey a status of command-waiting (or two stage command status) to any authorized command source
- The ability to relate special inhibits of functions to a managed object

F.2 SSFP Managed Object Attributes and Relationship Attributes

In order to minimize the complexity of its interfaces and the overall processor loading, the ISE needs to define a standard set of on-board managed object attributes. International standards for the management of open systems define attributes for the state, status, and relationship attributes of managed objects. The international standard state and status attributes of managed objects are the operational states, usage states, and administrative states and the procedural status, alarm status, availability status, and control status attributes.

A set of operational, usage, procedural, alarm status, availability status, and control status attributes is defined in international standard ISO/IEC 10164-2. The proposed relationship attributes for *Freedom* are modeled as one-way and two-way relationships as defined in international standard ISO/IEC 10164-3. Many of the proposed and relationship attributes are defined in international standards ISO/IEC 10164-2, and 10164-3.

F.2.1 Object Attributes

The following state and status attributes are defined in international standard ISO/IEC 10164. These state and status attributes may be used as appropriate for each on-board managed object class:

- The **operationalState**⁵⁰ attribute to indicate that the managed object is *enabled*⁵¹ (initialized) and the managed object is *disabled* (terminated)
- The **usageState** attribute to indicate the function of the managed object. The usage state includes *idle* (to indicate no current usage), *active* (to indicate current usage with spare capacity), and *busy* (to indicate current usage without spare capacity)
- The **administrativeState** attribute to indicate the Tier 1 administrative control of the managed object. The administrativeState has three attribute values. These three values are called *locked*, *unlocked*, and *shuttingDown* and are described further in ISO/IEC 10164-2, clause 7.1.3.
- The **alarmStatus** attribute to indicate the managed object's alarm status. The alarmStatus has the following possible attribute values:
 - **UnderRepair**: The object is currently being repaired. When the **underRepair** value is present, the operational state of the managed object is *disabled* or *enabled*.
 - **AlarmOutstanding**: One or more FDIR alarm messages with probable cause indicating a fault has been reported for the managed object and has not been cleared. (Note: alarm messages are defined in ISO/IEC 10164-4.) These faults may or may not have been disabling. If the operational state is enabled, additional attributes particular to the object class, such as built-in test results, indicate the services that are affected and the nature of the fault.
 - **Critical**: The **critical** severity level indicates immediate corrective action is required.

⁵⁰ In this document, the international method of creating object class, attribute type, and attribute value names is used. To make a name, descriptive phrases are concatenated and capital letters start each word in the concatenation.

⁵¹ In this document, the attribute values of the state attributes are italic.

- **Major:** The **major** severity level indicates urgent corrective action is required.
- **Minor:** The **minor** severity level indicates corrective action should be taken to prevent serious failure.
- The **proceduralStatus** attribute to indicate the run-time envelope supporting the initialization and termination of *Freedom* managed objects. This includes the initial start-up of the station, the start-up of partial assemblies, the start-up from a shut down, the sectional start-up from a component failure, the sectional start-up from sectional upgrades of hardware or software, the recovery from power outages, and the activation of an off-line or standby unit as part of FDIR. If empty, then none of the following conditions is present. It can have one or more of the following values, not all of which are applicable to every class of managed object:
 - **InitializationRequired:** The managed object requires initialization before it can be available for use, and this procedure has not been initiated. The manager (Tier 1) may be able to invoke such initialization through an action command. The operational state is *disabled*.
 - **Initializing:** The managed object requires initialization before it can be available for use, and this procedure has been initiated but is not yet complete. When the condition is present, the initialization required condition is absent since initialization has already begun. The operational state is *disabled*.
 - **Reporting:** The managed object is in the process of reporting (generating a notification as part of its predefined managed object behaviour). When the condition is present, the operational state is *enabled*.
 - **Terminating:** The managed object is in the process of transiting to the dormant state. When the condition is present, the operational state is *enabled*.
- The **availabilityStatus** attribute to support the determination of the managed objects. If empty, then none of the following conditions is present. It can have one or more of the following values, not all of which are applicable to every class of managed object onboard the space station:
 - **InTest:** The managed object is undergoing a test procedure. If the administrative state is locked or shutting down, then normal users are

precluded from using the object and the control status attribute has the value **reservedForTest**. Tests that do not exclude additional use of the object do not require the establishment of the **reservedForTest** value in the control status attribute.

- **Failed:** The managed object has a fault that prevents the object from operating correctly. The failure has been detected by an internal check, as opposed to human speculation. The operational state is *disabled*.
- **PowerOff:** The managed object requires power to be applied and is not powered on. For example, a standby unit that has not failed. The operational state is disabled.
- **OffLine:** The managed object requires some switching operation (sequence of commands) to be performed to make it available for use. The switching operation may be manual or automatic or both. The operational state is object *disabled*. (The off-line attribute could help determine the station dormant mode. In combination with the **powerOff** state, the station dormant status is determined.)
- **OffDuty:** The managed object has been made unavailable in accordance with an on-board operating plan. Some command and control processes within the command and control structure have taken the managed object out of service at a scheduled time. The operational state is *enabled* or *disabled*.
- **Dependency:** The managed object cannot operate because some other resource on which it depends is *disabled*. For example, a device is not accessible because its controller is poweredOff. The operational state is *disabled*.
- **Degraded:** The managed object has degraded in service, such as in speed or operating capacity. Failure of test or an unacceptable performance measurement has established that some or all services are not functional or are degraded due to the presence of a defect, fault, or error. However, the managed object remains available for service, either because some services are satisfactory or because degraded service is preferable to no service at all. Object specific attributes may be defined to represent further information. For example, the failure isolation and recovery services may indicate which services are not functional. The operational state is *enabled*.

- **NotInstalled:** The managed object is not installed or is incompletely installed. For example, a plug-in module is missing or a cable is disconnected. The operational state is *disabled*.
- **LogFull:** The managed object class of log is reporting a log full condition indicating that the managed object class instance is not available.
- The **controlStatus** attribute is the set of attributes that support the operations for management service controls. These attributes are related to command inhibits, command constraints, command overrides, command interlocks, and command procedures. If empty, then none of the following conditions is present. It can have one or more of the following values, not all of which are applicable to every class of managed object:
 - **SubjectToTest:** The managed object available to normal users but tests may be conducted. The since the object is available to normal users its administrative state is *unlocked*.
 - **PartOfServicesLocked:** This value indicates whether a Tier 1 inhibit has administratively restricted a particular part of a service from the users of the managed object (system, element, or payload). Examples are command constraints, or outgoing message discriminators on FDIR reports or TOLs. The administrative state is *unlocked*.
 - **ReservedForTest:** The managed object has been made administratively unavailable to normal users because it is undergoing a test procedure. The administrative state is *locked*.
 - **Suspended:** Service has been administratively suspended to the users of the resource. The administrative state is *unlocked*.

F. 2.2 Relationship Attributes

This section includes relationship attributes for use in describing the configuration of the space station systems that interface with ISE. These relationship attributes may be adopted as appropriate for each on-board managed object class.

- The **functionInhibit** attribute is used in a managed object definition to identify the list of functions that can be inhibited. The **functionInhibit** attribute is not defined in international standards. The function inhibit attribute is an array with two sub-

attributes for each function that may be inhibited. The first sub-attribute is the data object identification of the function. The second sub-attribute is the current state of the access control of this command (i.e., function is inhibited or uninhibited). The function inhibit attribute is set-valued and read-write. The functions of the managed objects that can be inhibited can be considered modes, or groups of command constraints depending on the designer's implementation of the behaviour (the software application) of the managed object.

- The **RPCObject** attribute is used in an on-board managed object definition to identify the remote power controller (RPC) supplying power to the managed object. The **RPCObject** attribute is not defined in international standards. The **RPC** attribute is a list of managed object identifiers and corresponding RPC. In the ISE, the attribute contains all of the RPC and the corresponding managed object connected to each. The attribute is set-valued and read-write.
- The **backupObject** attribute can be included in defined managed objects to identify a managed object acting in a back-up role with respect to it. The **backupObject** attribute is defined in international standard ISO/IEC 10164-3. The back-up object attribute is single-valued and read-only, although its value is NULL if the managed object that owns the attribute is currently active and not in need of back-up service. The back-up object attribute forms the back-up object parameter defined in the alarm reporting function standard (ISO/IEC 10164-4). The ISE assumes that each managed object would contain this attribute if it were capable of being backed up.
- The **backedUpObject** attribute can be included in defined managed objects to identify a managed object acting in a backed-up role with respect to it. The **backedupObject** attribute is defined in international standard ISO/IEC 10164-3. The backed up object attribute is single-valued and read-only, although its value is null if the managed object that owns the attribute is not currently active as a back-up on behalf of any other object. The ISE assumes that each managed object would contain this attribute if it were capable of being backed up.
- The **stationMode** attribute (a member object attribute) is proposed to indicate the relationship to each station mode. The **stationMode** attribute is not defined in international standards. There would be values of this attribute equal to the names of the station modes. The station mode attribute would be used for access control of the capabilities of the managed object. The member object attribute is set-valued and read-write.

- The **commandWaiting** status attribute is part of the command structure that allows any managed object to report that the action requested is waiting a precondition. The **commandWaiting** attribute is not defined in international standards. The **commandWaiting** attribute is a way to communicate to the managing system preconditions for an action that is currently pending in the managed object. For example, some contained managed object is not in the correct state. The managed object is waiting for the state to change or it is waiting for an action that will result in the correct state. The **commandWaiting** attribute is multiple valued and contains the **commandID** that is waiting.
- The **standbyStatus** attribute is used only if a back-up relationship exists. The **standbyStatus** attribute indicates if the back-up managed object is a hot standby, a cold standby or providing service. The attribute is single-valued and read-only. The standby status attribute has the following attribute values:
 - **ProvidingService**: The back-up resource is providing service and is backing up another resource. The providing service condition is mutually exclusive with the hot standby and cold standby conditions.
 - **HotStandby**: The resource is not providing service but is operating in synchrony with another resource that is to be backed-up (e.g., a computer shadowing another computer). A resource with a hot standby status will be immediately able to take over the role of the resource to be backed-up, without the need for initialization activity, and will contain the same information as the resource to be backed-up. The **hotStandby** condition is mutually exclusive of the **coldStandby** and **providingService** conditions.
 - **ColdStandby**: The resource is to back-up another resource, but is not synchronized with that resource. A resource with a cold standby status will not be immediately able to take over the role of a resource to be backed up, and will require some initialization activity.

F.3 The Attribute GDMO

The on-board object class is used to define the criteria for controlling management service. The following description follows the requirements of ISO international standard ISO/IEC 10165 part 4 -- Generic Definition of Managed Objects (GDMO). The attributes described are optional and are to be included only as necessary in the definitions of the object classes.

SSFPManagedObject MANAGED OBJECT CLASS

DERIVED FROM ---the space station's top object class: root;

CHARACTERIZED BY

ATTRIBUTES

operationalState	GET,
usageState	GET,
administrativeState	GET,
alarmStatus	GET,
proceduralStatus	GET,
availabilityStatus	GET,
controlStatus	GET,
standbyStatus	GET,
commandWaiting	GET,
functionInhibits	GET-REPLACE,
override	GET-REPLACE,
rpcObject	GET-REPLACE,
backupObject	GET-REPLACE,
backedUpObject	GET-REPLACE,
stationMode	GET-REPLACE;

Operational State

The operationalState attribute is specified in Recommendation X.731 ISO/IEC 10164-2, REGISTERED AS {smi2AttributeID 35}; -- The attribute is mapped to a DSSRDB.ATTRIBUTE_T.

Usage State

The usageState attribute is specified in Recommendation X.731 ISO/IEC 10164-2. REGISTERED AS {smi2AttributeID 39}; -- The attribute is mapped to a DSSRDB.ATTRIBUTE_T.

Administrative State

The semantics of the administrativeState attribute type are specified in the Recommendation X.731 ISO/IEC 10164-2.

REGISTERED AS {smi2AttributeID 31}; -- The attribute is mapped to a DSSRDB.ATTRIBUTE_T.

Alarm Status

The alarmStatus attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 32}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Procedural Status

The proceduralStatus attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 36}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Availability Status

The availabilityStatus attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 33}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Control Status

The control status attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 34}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Standby Status

The standby status attribute is specified in Recommendation X.731 ISO/IEC 10164-2.
REGISTERED AS {smi2AttributeID 37}; -- The attribute is mapped to a
DSSRDB.ATTRIBUTE_T.

Command Waiting

The commandWaiting attribute is the identification of the command that is waiting the
change of attribute values or states before the command is executed.

CommandWaiting ATTRIBUTE
WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.ISECommonAttributes;
MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

Function Inhibit

The functionInhibit attribute is the identification of the functions and the state of inhibit for each function in the managed object. This attribute is present only if the managed object has functions that needs to be constrained or is defined to have modes that are sets of constraints. The attribute values of the **inhibitState** of functionInhibit are *inhibited* or *uninhibited*.

```
functionInhibit ATTRIBUTE
    WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.ISECommonAttributes;
    MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};
```

RPC Object

The RPCObject attribute is the identification of the connection between the managed object and its RPC.

```
RPCObject ATTRIBUTE
    WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.ISECommonAttributes;
    MATCHES FOR Equality;
REGISTERED AS {DSSRDB.ATTRIBUTE: ??};
```

Backup Object

The backupObject attribute is the identification of the back-up managed object(s).

```
backUpObject ATTRIBUTE
    WITH ATTRIBUTE SYNTAX-Attribute-ASN1Module.BackUpRelaltionshipObject;
    MATCHES FOR Equality;
REGISTERED AS { smi2AttributeID 40};
```

Backed-Up Object

The backedUpObject attribute is the identification of the backed-up managed object(s).

```
backedUpObject ATTRIBUTE
    WITH ATTRIBUTE SYNTAX-Attribute-ASN1Module.BackUpRelaltionshipObject;
    MATCHES FOR Equality;
REGISTERED AS { smi2AttributeID 41};
```

Station Mode

The stationMode attribute is the identification of the station modes that the managed object supports.

stationMode ATTRIBUTE

WITH ATTRIBUTE SYNTAX-Attribute-DMSModule.ISECommonAttributes;

MATCHES FOR Equality;

REGISTERED AS {DSSRDB.ATTRIBUTE: ??};

DMSModule.ISECommonAttributes

CommandWaiting::= SET OF {CommandID}

FunctionInhibit::=SET OF {Inhibits}

Inhibits::	SEQUENCE {	
	functionID	ObjectID
	functionInhibitState	InhibitState}

InhibitState::= ENUMERATED ----[1] inhibited, [2] enabled

RPCObject::= SET OF{RPCControllers-}

RPCControllers-::= SEQUENCE {	
	objectID
	rpcID
	DistinguishedName
	DistinguishedName}

StationMode::= SET OF{	
	systemMode
	Integer}

APPENDIX G

LOG CONTROL FUNCTION

This section of the document describes the log⁵² control function that may be used by application processes in the Space Station *Freedom* Program. The log control function is proposed to meet the requirements of the SSFP Tier 1 to monitor and control the logging of log records⁵³ representing commands, reported event notifications, TOLs (reported summarization reports), security breach notifications, all alarm notifications, and all FDIR notifications. The ISE as part of Tier 1 needs a flexible logging report control service that allows ISE to preserve information about commands and events that may have occurred or operations that may have been performed by or on various objects. In the on-board systems various DMS resources (Mass Storage Unit, SDP memory, NOS memory, etc.) may store such information. The Tier 1 components (ISE, SSCC, and POIC) will model these resources as logs and log records contained in the logs. The type of information that is to be logged may change from time to time, and furthermore, when such information is retrieved from a log, the Tier 1 component must be able to determine whether any records were lost or whether the characteristics of the records stored in the log were modified at any time.

Thus, Tier 1 needs a flexible log control service that allows selection of records to a particular log. Tier 1 needs the ability to modify the criteria used in logging records. Tier 1 needs the ability to determine whether the logging characteristics were modified or whether log records have been lost. Tier 1 needs a mechanism to control the time during which logging occurs, (for example, suspending logging of TOLs during times when the space network is communicating to the SSCC and resuming logging of selected TOLs during the zone of exclusion [ZOE]). Tier 1 needs the ability to retrieve and delete selected log records. And Tier 1 needs the ability to "initialize and enable" (create) logs and "shutdown, terminate, make dormant" (delete) logs.

A standard log control management function that provides these basic needs would provide a systematic and flexible command structure. The following sections include description of the log control function standardized by ISO/IEC. This log control function meets the needs of the Tier 1 components. Section 5 of this document includes findings, recommendations, trades, and risks associated with this design of a standard log control function.

⁵² Log: A log is a management support object class that models resources used as a repository for log records.

⁵³ Log record: A log record is a management support object class that models units of information stored in a log.

G.1 The Model of the Log Control Function

Each *Freedom* object needs a DMS STSV for the logging of event notifications and commands. The objects, systems, elements, and payloads with their attributes and their event notifications will send messages to Tier 1 components. Some of these messages will be notifications that are stored in object defined log records. The objects, systems, elements, and payloads are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP 30261]). This data standard refers to an applicable document, the SMI, ISO/IEC 10165. SMI part 2 contains the ISO attributes and objects used for managing log controls as described in ISO/IEC 10164-6, *Information Processing Systems - Open System Interconnection - System Management - Part 6: Log Control Function*. The ISO/IEC IS 10164-6 standard provides a standard way of managing logs. The SMI standard (ISO, 1991 [10165]) defines syntax for the attributes, objects, and the generic notification of the log control function. The ISO/IEC 10164-6 standard describes how the attributes, and objects of the event manage function work together to provide the log control function. The ISO/IEC standard 10164-6 provides detail on the attributes and objects for the log control function, and it consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), the *Open System Management Overview* (ISO, 1991 [10040]), and the other parts of ISO/IEC 10164.

The standard ISO/IEC 10164-6 specifies the log control function services as a generic storage resource that stores copies of information and is controllable with log control commands. The log is a repository for records. The records contain logged information. Information to be logged is obtained from reported event notifications, object management notifications, state management change notifications, relationship change notifications, commands, C&W alarm notifications, FDIR alarms, security breach notifications, and communication protocol data units. The log object provides the generic storage resource that is controlled with log control commands.

The log object class is characterized by ISO/IEC 10165-6 as having the following attributes:

- A log identifier, uniquely identifying an instance of a log in terms of where it is contained
- An administrative state, operational state, and availability status, representing the object states of the log
- The time during which logging is active (This attribute is supported by the conditional scheduling packages. See section 4.5, The Event Control Function.)

- A description of the type of information to be logged (This attribute is supported by the discriminator construct attribute. See section 4.5.)
- The maximum log size
- The current log size
- The number of records currently in the log (Together with the current log size this is used to obtain an estimate of the average record size, and therefore, the number of records that can still be logged.)
- The log full behaviour when its maximum capacity is reached
- The capacity alarm thresholds defined as percentages of the maximum log size (These capacity alarm thresholds are used to generate events that indicate various levels of the log full conditions. This property is supported by the threshold attribute.)

The log object also has the object management notifications that are generated when the log is initialized and enabled (created); shutdown, disabled, and made dormant (deleted); suspended; resumed; and modified. (This property is supported by the object creation, object deletion, state change, and attribute change notification of ISO/IEC 10164-1 and ISO/IEC 10164-2.)

The ISO/IEC 10165-6 standard specifies the behaviour of the log. The log behaviour is determined by its state attribute, availability status, scheduling packages, and its discriminator construct. The behaviour of the log is characterized by the following rules:

- The log stores records in the order in which they are presented for logging. New records that pass the discriminator construct test will only be stored if the log is in the unlocked administrative state and is not in the enabled and log-full (for a log that halts), disabled, or off-duty state.
- When a log is in the "locked" administrative state, the log will not store new records and records contained in the log are available for retrieval. When the log is in the "unlocked" administrative state, records currently contained in the log can be retrieved unless the log is in the "disabled" operational state.
- The log operational state cannot be changed by direct command action, but reflects the internal activity of the log. For the behaviour of the log, when the maximum

log size has been reached (the log-full availability status), two options are defined. The log may either halt logging or the log may wrap. A log that halts upon reaching the log-full condition will always generate a capacity threshold notification that indicates that this condition has been reached. The behaviour of such a log corresponds to a log that discards new information in preference to older information. A log that wraps upon reaching the log full condition will discard an integral number of records in order to log new records. The log may also generate a capacity threshold event notification indicating that a new wrap has occurred. The wrap log corresponds to a log that discards old information in preference to new information. Every log must be able to support the halt type behaviour; support for the wrap behaviour is optional.

The management of the log is through the modification of the log object's class attribute values, though restrictions do exist. For example, the maximum log size attribute may not be modified to a value less than the current size of the log. Whenever an attribute is modified (via set command like DMS-WRITE), an attribute change notification is generated. This notification, depending on the assertions of its corresponding event forwarding discriminators, will be sent to the Tier 1 components identified by the event forwarding discriminators. To verify and determine the status of the accuracy of information contained in a log, the discriminator passes events for the log identified. This is done by reading (via DMS-READ) the discriminator construct in the logging system (history standard service). Once the log record objects pertaining to the log have been identified, the history of the log can be reconstructed.

The standard specifies the log control function's mandatory attributes as follows:

- The **log ID** identifies the instance of a log object.
- The **discriminator construct** tests the information that is to be logged. The discriminator construct operates on any of the parameters (the fields of the notifications or messages) of the information to be logged.
- The **administrative state** represents the capability of the log to function. The standard defines the following administrative states:
 - The log **unlocked** state is for logging and retrieval of records
 - The log **locked** state makes the log unavailable for logging of new records.
- The **availability status** qualifies the operational state of the log. The attribute may indicate a "log-full" condition.

- The **maximum log size** specifies the size of the log measured in octets. A log may be of an indeterminate size. A maximum log size of zero specifies that the log size has no predefined limit.
- The **current log size** specifies the current size of the log measured in octets and is based upon the actual amount of information that is contained in the data representation used in the log. The log size does not include any overhead. Immediately after the initialization and log enable (creation), the current log size should be zero.
- The **number of records** specifies the current records in the log.
- The **capacity alarm threshold** specifies, as a percentage of the maximum log size, the values at which a threshold event notification will be generated to indicate either a full log or log wrap condition. The attribute is set-valued (via DMS-WRITE). The attribute must support the halt behaviour. When a log has the wrap option, the capacity threshold events are triggered as if coupled to a gauge that counts from zero to the highest capacity threshold value defined and then resets to zero.
- The **log full action** specifies what behaviour is selected when the log reaches its maximum size. The value of the attribute is wrap or halt.

The standard specifies the three conditional scheduling packages for a log instance. The scheduling packages are the **daily scheduling package**, the **weekly scheduling package**, and the **external scheduler scheduling package**. These scheduling packages are the same as those specified for the event report function (see section 4.5).

The standard specifies that log records are managed as objects that represent information stored in logs. The log record managed object class serves as a superclass for other record classes. As part of the specialization of the log record class, additional attributes may be assigned to the new subclass. The log record class has the following properties:

- A log record identifier
- A logging time
- An object creation notification
- An object deletion notification

The standard specifies a log record behaviour as follows:

- Log records are created as a result of an event notification or side-effect of some management operation, they are not created explicitly by management operations or commands.
- Log records may be retrieved and deleted; the attributes of a log record cannot be modified.
- The operations performed on a log record depend on the state of the log in which the records are contained and may also be subject to access control security constraints (e.g., command inhibits).

The standard specifies the log record mandatory attributes as follows:

- The log record ID identifies each record in the log. The log record identifier is a number that is unique within the scope of the log and is assigned sequentially. The identification number used may wrap, however at no time shall there be more than one record with the same identifier in the log. The log record ID is an integer.
- The logging time contains the value of the time when the record was entered into the log. (Note that event notifications have a time parameter that indicates the time that the event occurred.) The contents (parameters) of an event log record is determined by the event notification type. To allow the retrieval of attributes from logs, attributeIDs are assigned in all notification templates⁵⁴.

G.2 Notifications of the Log Control Function

The log control function uses event notifications to report changes in the log control function. The standard specifies five log control function notifications. The first four used notifications are as specified by ISO/IEC 10164-1, Object Management Function. The last is specified by ISO/IEC 10164-4, Alarm Reporting Function.

- State change notification
- Attribute value change notification

⁵⁴ Template: Templates are the defined formats or forms for various object oriented properties. The ISO/IEC 10165-4 standard defines the templates, for notification, objects, attributes, behaviour, etc. Thus, ISO/IEC 10165-4 tells object definers how to "spell" and how to "write and generate standard information" about the various properties of objects.

- Create notification
- Delete notification
- Alarm notification

In reporting the capacity threshold event, the alarm notification is used. The standard for log control function specifies the following use of the alarm notification parameters:

- The **managedObjectClass** parameter identifies the log class.
- The **managedObjectInstance** parameter identifies the instance of the log generating the notification.
- The **alarmType** parameter indicates that a processing error has occurred.
- The **severity** parameter indicates the severity assigned to the capacity threshold event. When the 100% log full condition is reached, a severity value of critical is assigned to the event.
- The **monitoredAttributes** parameter carries the maximum log size attribute value.
- The **probableCause** parameter carries the attribute value of the **storageCapacityProblem**.
- The **thresholdInfo** parameter carries the capacity threshold value (as a percentage of total capacity that was reached or exceeded in generating this event.)

The ISO/IEC IS 10164-1, clause 11, and ISO/IEC 10164-6, clause 8, provide the mapping of the parameters of the notifications to the CMIS parameters.

G.3 Attributes and Objects for Log Control Function Service Definitions

The attributes and objects for representing the log control function will be provided by the detail design of the DMS, ISE and the *Freedom's* objects, systems, elements, and payloads. The DMS STSV should have a log control function to meet the needs of the SSFP. DMS STSV should provide the objects and attributes of the standardized log control function. Examples of how DMS could provide the log control functions are provided in the ISO/IEC IS 10164-6. The designs of the ISE and DMS do not have to comply with ISO/IEC IS 10164-6, but the capability of DMS will require the functions of the standard.

The attributes for the log control function are defined and explained by the ISO/IEC IS 10164-6. The ISO/IEC IS 10165-2 defines the abstract syntax for the log control function attributes.

The ISO/IEC IS 10164-6 specifies the following attributes for the log function:

- logID
- discriminatorConstruct
- availabilityStatus
- administrativeState
- operationalState
- StopTime
- StartTime
- weekMask
- schedulerName
- objectClass
- objectInstance
- maxLogSize
- currentLogSize
- numberOfRecords
- capacityAlarmThreshold
- logFullAction
- logRecordID
- LoggingTime

The ISO/IEC 10164-6 standard specifies service for manipulation of the log managed object class. These services consist of the following:

- Initiation of logging
- Termination of a logging
- Modification of log attributes
- Suspension of logging
- Deletion and retrieval of log records
- Resumption of logging

Thus, these services provide the means for Tier 1 to initiate, terminate, suspend, resume, and modify the logging capability.

G.4 Log Control Function Protocol and Abstract Syntax Definitions

The ISO/IEC 10165-2 specification defines the ASN.1 value notations for all the objects and attributes needed by the log control function.

APPENDIX H

SUMMARIZATION FUNCTION (Telemetry Object List Management)

This section of the document describes the objects and attributes for the summarization function⁵⁵ that may be used by the application processes in the Space Station *Freedom* Program. The summarization function is proposed to meet the requirements of the SSFP Tier 1 to obtain summary information from the observed attributes⁵⁶ of a managed object (system, element, or payload). The ISE as part of Tier 1 needs a flexible summarization function that allows Tier 1 to monitor the performance of the station. The SSFP Tier 1 components also needs to have a consistent set of definitions and actions related to management of the summarization function. In cooperation with the *Freedom* object management function (see section 4.1) and state management function (see section 4.2), event reporting function, and the service control functions, Tier 1 needs the ability to manage the summarization function.

Each *Freedom* object needs a DMS STSV for performance monitoring. The objects, systems, elements, and payloads with their attributes and their event notifications require periodic scanning, measuring, and reporting of a preselected set of observed attributes. Specific controls are required of the periodic scanning, measuring and reporting of the observed system, element, or payload attributes.

The Tier 1 components need performance monitoring to measure throughput, response times, availability, and other measures of congestion and system, element, or payload utilization. The summarization function needs to provide for the following user needs:

- The ability to report individual attribute values or derive attribute values such as aggregates and statistical information about the attribute values of objects, systems, elements, or payloads. (For example, scanning a list of voltage sensors and either

⁵⁵ Summarization: A summarization is the process of gathering and optionally applying algorithms to raw or observable information to produce summary information.

⁵⁶ Observed attribute: Observed attribute is an attribute of a managed object, system, element, or payload whose value is being observed by a metric object or a summarization object. (For example, all attributes in the RODB are observable. Any derived values from the attribute values in the RODB are provided by a metric object. The DMS STSVs scanning the RODB and generating TOLs are summarization objects.) .

reporting all the values or calculating and reporting a derived average and a standard deviation.)

- The ability to support the request for changes in the selection of attribute values reported. (For example, having a way to select and enable alternate TOLs.)
- The ability to summarize information on the following:
 - A single attribute type of a single managed object (For example, a running average of the bandwidth use of the S-band downlink communication channel. The ability to monitor uses of capacity is a special function that has its own detailed user needs (see ISO, 1991 [CD 10164-11], Workload Monitoring Function).
 - Multiple attribute types from a single managed object (For example, all the performance parameters of the secondary power distribution system.)
 - A single attribute type of multiple managed objects (For example, the electrical current levels detected for each SPC.)
 - Multiple attribute types from multiple objects. (For example, an abbreviated list of important attribute values from the core system for use during ZOE.)
- The ability to summarize information gathered:
 - At a single point in time, prescheduled or on-demand
 - Over a specified interval of time
 - Periodically over specific intervals of time
- The scheduling of summarization activity over a specified period of time. (For example, having special sets of sample sensors during station modes and during ZOE.)
- The ability to identify and relate the summarized attribute and the corresponding managed objects
- The identification of any algorithms used for calculating statistical measures
- The ability for a managed system to send event notifications to the Tier 1 components to report:

- Any missing samples in the summary
 - The parameters of algorithms used for calculating any statistical measures (For example, the sampling frequency or the number of samples in a moving average.)
 - The identification of the sources of the attributes used in the summary
 - The identification of the start and stop times of the summary period
 - The identification of the units of measures
 - The summarization result (For example, the list of observed and derived values)
- The ability to provide efficient reporting of large quantities of summarized information. (For example, the structuring of the parameters of the notifications so that the position of the attribute value in the list is always the same.)
 - The ability to efficiently select objects whose attributes are to be summarized. (For example, summarize the core system applications use of the mass storage unit.)
 - The ability to optionally timestamp the observed values.

Standard performance summarization of objects and attributes that provides these basic needs would form a part of a systematic and flexible monitoring and status structure. The following sections include a description of the summarization function proposed for standardization by the committee draft, ISO/IEC 10164-13. This function identifies managed objects, their attributes, and their notifications (TOLs) that meet the needs of the Tier 1 components. Section 5 of this document includes findings, recommendations, tradeoffs, and risks associated with supplying a DMS STSV with this design of objects, attributes, and event notifications.

H.1 The Model of the Summarization Function

The objects, systems, elements, and payloads are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP30261]). This data standard refers to an applicable document, SMI ISO/IEC 10165. SMI part 2 will contain the ISO attributes and objects used for the performance monitoring

functions when the following committee drafts become international standards: the summarization function, ISO/IEC CD 10165-13, and the workload monitoring function, ISO/IEC CD 10164-11. The model of the summarization function's objects has become stable. The management service control attributes are likely to change. The basic attributes are unlikely to change, and besides only a selected set of summarization objects and basic attributes are needed for the specification of a standard DMS service for providing a summarization function (TOL services). Currently, the committee draft ISO/IEC CD 10164-13 defines syntax for the attributes and objects, and the generic notification of the objects and attributes for the summarization function. This draft also describes how metric objects and their attributes and event notifications work together to meet the Tier 1 summarization needs. The ISO/IEC committee draft 10164-13 also consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), the *Open System Management Overview* (ISO, 1991 [10040]), and the other parts of ISO/IEC 10164.

The committee draft ISO/IEC 10164-13 specifies the use of metric objects⁵⁷, metric attributes⁵⁸ as defined in ISO/IEC 10164-11, and log records as defined in ISO/IEC 10164-6. The committee draft specifies that summarization objects obtain information, process such information to produce information, and then issue summary notifications. Figure 19 illustrates summarization objects observing attributes within observed objects⁵⁹. The summarization object generates summaries (TOLs) in the form of notifications according to a specified reporting schedule or as the result of a request. The notifications may be forwarded as event notifications by event forwarding discriminators (see standard ISO/IEC 10164-6). The notification (TOL) may also be logged (see ISO/IEC 10164-6).

The summarization object instance includes attributes for schedules to control the underlying scanning and summary reporting process and various mechanisms to select observed objects and their attributes to be observed.

⁵⁷ Metric object: A metric object is defined in ISO/IEC 10164-11 as a managed object that contains at least one attribute whose value is calculated from the values of attributes observed in managed objects.

⁵⁸ Metric attribute: A metric attribute is defined in ISO/IEC 10164-11 as an attribute if a metric object whose value is either used as a parameter of one or more metric algorithms or whose value represents the output of such an algorithm.

⁵⁹ Observed object: An observed object is a managed object with attribute values that are observed by a metric object or a summarization object.

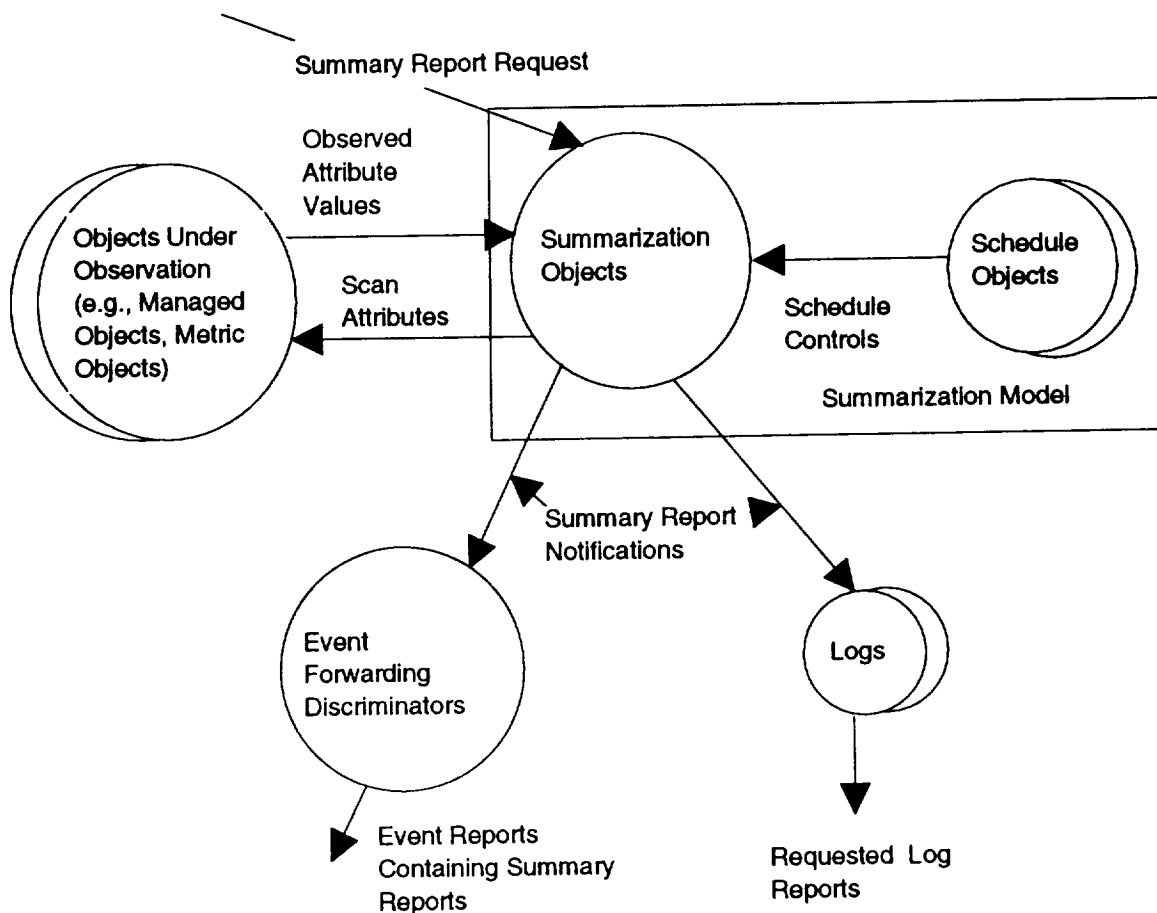


Figure 19. Summarization Objects Observing Attributes Within Objects Under Observation

The committee draft, ISO/IEC 10164-13 provides three object classes defined for summarization objects. They are derived from the **scanner** object class defined in the committee draft. These three summarization objects are as follows:

- The **homogeneous scanner**⁶⁰ whose behaviour is to scan⁶¹ a common set of arbitrary attribute types across managed objects selected using either a scoping and

⁶⁰ Homogeneous scanner: A homogeneous scanner is a scanner that collects values from the attributes of the same type from one or more managed objects. (For example, the scanning of temperature values from temperature sensors.)

filtering mechanism or a specified list of managed objects. In addition, the object class optionally calculates statistics for numeric attributes⁶² over a specified collection period⁶³ across the selected objects. At the end of each report period,⁶⁴ it reports either:

- The aggregate data collected
 - The statistics it has calculated
 - Or both the aggregate data and the statistics.
- The **heterogeneous scanner**⁶⁵, whose behaviour is to scan potentially different sets of attributes for a set of explicitly named observed objects and reports the results at the end of each granularity period⁶⁶ (scan).
 - The **heterogeneous buffered scanner**, whose behaviour is similar to the heterogeneous scanner, but stores scanned values so that the results of multiple granularity periods (scans) can be reported together. In addition, at the time of reporting, it will scan a list of attributes of arbitrary type whose attribute values can also be included in the notification.

⁶¹ Scan: A scan is a sampling process of observing attribute values at a specified point in time.

⁶² Numeric attribute: A numeric attribute is an attribute whose value may be either an integer (or possibly treated as an integer) or real number.

⁶³ Collection period: A collection period is the time during which a metric algorithm is applied to observe data. The collection period includes both the scanning and the calculating.

⁶⁴ Report period: The report period is the time between emitting notifications containing the collected aggregate values or statistical information.

⁶⁵ Heterogeneous scanner: A heterogeneous scanner is a scanner that collects values from potentially different sets of attributes for a set of explicitly named observed object instances and report the results at the end of each scan.

⁶⁶ Granularity period: Granularity period as defined in ISO/IEC 10164-11 is the time between observations of a managed object. In the context of ISO/IEC 10164-13, granularity is the time between the initiation of two successive scans.

The **scanner** object class is a superclass from which all other summarization objects are derived. The committee draft defines its facilities for periodically sampling the values of a specified set of attributes within specified objects. The intervals during which the periodic scans may occur can be controlled according to a schedule. The scanner object class has the following attributes:

- **ScannerID** whose value identifies an instance of the scanner object class (used for naming)
- An **operational** state as defined in ISO/IEC 10164-2 (see section 4.2)
- An **administrative** state defined in ISO/IEC 10164-2
- An **availability** status as defined in ISO/IEC 10164-2
- The **granularity** period indicating the time between scans
- **Scheduling** attributes defined in ISO/IEC 10164-5 (see section 4.6)

All the object classes defined by the committee drafts have some common behaviour related to generating notifications on demand and time stamping observed attribute values. If scheduling has zero reporting intervals, then the scanners report their notification on the receipt of an action command to generate the notification. Also, a scanner can optionally report timestamp with the observed values.

The **homogeneous scanner** has the following attributes in addition to those inherited from scanner.

- Basic attributes included in all homogeneous scanners are:
 - The **scan attribute identifier list** attribute identifies the attributes whose values are to be included in the notification for each of the selected objects. Each attribute value reported is paired with its attribute identifier.
 - The **numeric attribute identifier array** attribute identifies an ordered sequence of numeric attributes whose values are to be included in the notification for each of the selected objects. The numeric attribute values are reported as a sequence in the order of the attributes specified in the numeric attribute identifier array without the attribute identifiers.

- A conditional **scoped selection package** contains the following attributes:
 - The **base object identifier attribute** includes both the class and instance identifier of the base managed object used for scoping (as defined in ISO/IEC 9595).
 - The **discriminator construct** attribute specifies the criteria to be used for selecting objects (as defined in ISO/IEC 10164-5).
 - The **scope** attribute specifies the levels in the naming hierarchy which identifies the instances that are to be checked using the discriminator construct, the time attribute identifier, and time offsets.
- A conditional **timing selection package** that is only present if the scoped selection package is also present, contains the following attributes:
 - The **begin time offset** attribute produces a time window relative to the current time when used with the attribute **end time offset**. The value of the attribute is subtracted from the value of current time to define the beginning of a time window.
 - The **end time offset** attribute is subtracted from the value of current time to define the end of a time window.
 - The **time attribute identifier** whose value is an object identifier that points to another time attribute whose values have the syntax of ASN.1 type GeneralizedTime. The time values are used in the comparison against the time window in order to select managed objects (log records).
- A conditional **managed object instance selection package** contains the **object list** attribute that identifies the set of object instances whose common attributes are to be observed for the notification. The managed object instance selection package and the scoped selection package may **NOT** be both present.
- A conditional **simple scanner package** that generates the notification of the observed attribute values at the end of each scan if the summarization does not require the calculation of any values from the collected attribute values.
- A conditional **statistical package** that observes attributes and calculates values at the end of each collection period and has the following attributes:
 - An **algorithm type** attribute, which identifies the algorithm to be used in the calculation of the statistics of the common attributes. Three algorithm types

- are specified in ISO/IEC 10164-11. The algorithm types are the sample mean, the sample mean and variance, and the sample mean and percentiles.
- The **collection period** attribute, which specifies the time period during which scanning and calculating occur.
- A conditional **scheduling package** includes the attributes as defined in ISO/IEC 10165-5. (see section 4.6)

The **heterogeneous scanner** has the **observation identifier list** attribute, in addition to those inherited from the scanner. The **observation identifier list** attribute is a set of object classes, object instances, and associated attribute identifiers. The attributes to be observed in each of the selected objects may be specified in either, or both, of two lists included in the observation identifier list. The list used to identify the attributes are the **scan attribute identifier list** and the **numeric attribute identifier array**.

The **heterogeneous buffered scanner** has the following attributes, in addition to those inherited from the scanner:

- A report period attribute
- Scheduling package's attributes that control the times during which it does periodic reporting
- A buffered observation identifier list attribute, which identifies each instance of managed object attributes. The buffered observation identifier list is organized by managed object and for each managed object there are three attribute identifier lists that may be specified. An attribute identifier may be present in more than one of these lists. The three lists are:
 - The scan attribute identifier list attribute, which specifies attributes of any type.
 - The numeric attribute identifier array attribute, which specifies an ordered sequence of numeric attributes.
 - The report time attribute identifier list attribute, which specifies a set of attributes of arbitrary types that are to be scanned at the end of each report period and is included only once in the notification.

H.2 Notifications of the Summarization Function

The committee draft defines six notifications of the summarization function:

- The homogeneous scan notification
- The statistic summary notification
- The heterogeneous scan notification
- The heterogeneous buffered scan notification
- The activate scan notification
- The activate buffered notification

The committee draft maps the parameters of the notification to either the CMIS event report service or the CMIS action service. The first four are mapped to the CMIS event report service, and the last two are mapped to the CMIS action service. The committee draft in clause 11 maps the notification parameters to the CMIS parameters.

Also the committee draft specifies the use of services to modify the operation of the summarization objects. In particular, these specified operations use the following standard notifications:

- Attribute value change notification
- Create notification
- Delete notification

The ISO/IEC IS 10164-1, clause 11, provides the mapping of the parameters of these notifications to the CMIS parameters.

H.3 Attributes and Objects for Objects and Attributes for Summarization Function Service Definitions

The attributes and objects for representing the summarization functions will be provided by the detail design of the DMS, ISE and the *Freedom's* objects, systems, elements, and payloads. The DMS STSV should have objects and attributes for a summarization function to meet the needs of the SSFP. Examples of how DMS could provide the objects and attributes for the summarization function are provided in the ISO/IEC CD 10164-11 and 13. The designs of the ISE and DMS do not have to comply with ISO/IEC CD 10164-11 or ISO/IEC 10164 -13, but the capability of DMS will require the functions of the standard.

The attributes for the objects and attributes for the summarization function are defined and explained by the ISO/IEC CD 10164-13.

The ISO/IEC CD 10165-13 specifies the following attributes for the summarization function:

- AlgorithmType
- BaseManagedObjectID
- CollectionPeriod
- ReportPeriod
- BeginTimeOffset
- EndTimeOffset
- NumericAttributeArray
- ObjectList
- BufferedObjectList
- ObservationIDList
- ReportIntervalsOfDay
- ReportSchedulerName
- ReportStartTime
- ReportStopTime
- ReportWeekMask
- ScanAttributeIDList
- ScannerID
- Scope
- TimeAttributeIdentifier

H.4 Protocol and Abstract Syntax Definitions of Objects and Attributes for The Summarization Function

The ISO/IEC 10165-13 committee draft defines the ASN.1 value notations for all the objects, attributes, notifications, actions (commands), and behaviours needed by the objects and attributes for the summarization function. These abstract syntax definitions will move to ISO/IEC 10165-2 when ISO/IEC 10164-13 becomes an international standard.

APPENDIX J

THE TEST MANAGEMENT FUNCTION

This section of the document describes the objects and attributes for the test⁶⁷ management function that may be used by application processes in the Space Station *Freedom* Program. The test management function is proposed to meet the requirements of the SSFP Tier 1 to obtain test information from the observed attributes⁶⁸ of a managed object (system, element, or payload). The ISE, as part of Tier 1, needs a flexible test management function that allows Tier 1 to perform confidence tests and built-in tests (BIT). The SSFP Tier 1 components also needs to have a consistent set of definitions and actions related to the management of the test management function. In cooperation with the *Freedom* object management function (see section 4.1), and state management function (see section 4.2), event reporting function (see section 4.5), and performance management using the summarization function (see section 4.8), Tier 1 needs the ability to manage the test management function.

Each *Freedom* object needs a DMS STSV for test management. The objects, systems, elements, and payloads with their attributes and their event notifications require on-demand and periodic testing capabilities. Specific controls are required of a test function that can perform the remote testing and reporting of the test results to the Tier 1 components. The test management function needs to provide for the following user needs:

- The ability to collect test data that may either be used in the diagnosis of failure, or in the routine gathering of performance statistics. (Note that the summarization function may report individual attribute values or derive attribute values useful to diagnosis of failures.)

⁶⁷ Test: A test is the operation and monitoring of an object, system, element, or payload within an environment designed to elicit information on the functions and/or the performance of the subject.

⁶⁸ Observed attribute: An observed attribute is an attribute of a managed object, system, element, or payload whose value is being observed by a metric object or a summarization object. (For example, all attributes in the RODB are observable. Any derived values from the attribute values in the RODB are provided by a metric object. The DMS STSVs that scans the RODB and generates telemetry object lists are summarization objects.) .

- The ability to create the environment for the test, the control and monitoring of the subject systems (i.e., operation of the test), and the reassertion of the normal environment. Control of a test includes the need to suspend and resume tests. Each test will require a unique identification so that, for example, data generated by the test can be tracked. Features of the systems environment that may require alteration for testing include the following:
 - The connections to the subject system (see object management function, section 4.1)
 - The configuration of the subject system (see state management function, section 4.2)
 - The measurement of the workloads requested of the subject system (see the summarization function, section 4.8).
- The ability to schedule tests. The scheduling of test start and test termination must be considered in both a periodic and an aperiodic way. There is also the need to conduct the test at a time convenient to the subject system. For such tests there is the requirement to allow modification of the schedule (see section 4.10, the scheduling function).
- The ability to initialize and enable (create) complex tests from simpler ones, for example, to provide the results of many subordinate tests into a single result, or to sequence tests to efficiently diagnose a failure in an entity with a large number of components.
- The ability to correlate the results of each test in order to formulate an outcome from a set of individual tests.
- The ability to apply the same test management function to different test methodologies, such as a loopback test that configures the subject system to return the data it receives, a built-in-test that simply provides a pass-fail indication, and an error injection test that verifies that the errors are handled properly.
- The ability to optionally timestamp the observed test values.

Standard test management objects and attributes that provides these basic needs would form a part of a systematic and flexible test monitoring and status structure. The following sections include descriptions of the test management function proposed as a standard by the committee draft, ISO/IEC 10164-12. This function identifies managed objects, their

attributes, and their event notifications that meet the needs of the Tier 1 components. Section 5 of this document includes tradeoffs, issues, and risks associated with supplying a DMS STSV with this design of objects, attributes, and event notifications.

J.1 The Model of the Test Management Function

The objects, systems, elements, and payloads are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP30261]). This data standard refers to an applicable document, the SMI ISO/IEC 10165. SMI part 2 will contain the ISO attributes and objects used for the test management functions, when the test management function, ISO/IEC CD 10164-12, becomes an international standard. The model of the test management function's objects is not very stable. The test management model has to pass many international ballots before it becomes stable, however, it does provide a design for a test management function. The Test Management Model could be used as a set of guidelines for the development of level C requirements. Currently the committee draft ISO/IEC CD 10164-12 defines syntax for the attributes, objects, and the generic notification of the objects and attributes for the test management function. This draft describes how testing objects and their attributes and their event notifications would work together to meet the Tier 1 test management needs. It also consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), the *Open System Management Overview* (ISO, 1991 [10040]), and the other parts of ISO/IEC 10164.

The committee draft ISO/IEC 10164-12 specifies the terminology concerned with tests. It describes and defines terms used to describe the test environment. It describes two test environments: the asynchronous test and the synchronous test. In both environments, the test conductor object⁶⁹ initiates the test from the managing process. The model for these test environments is illustrated in figure 20. A test performer object⁷⁰ conducts the test in the managed process. Figure 20 illustrates a single instance of a test invocation, showing only a subset of possible message exchanges. A test request is addressed to a managed object that has the ability to receive and respond to such requests. Such ability is called the test action

⁶⁹ Test conductor : A test conductor is a manager or managing object that issues test operations (i.e., commands).

⁷⁰ Test performer: A test performer is an agent (i. e., an object or system) that receives test operations (i.e., commands).

request receiver⁷¹ (TARR). (The ISE and a DMS STSV could provide this capability onboard *Freedom*.)

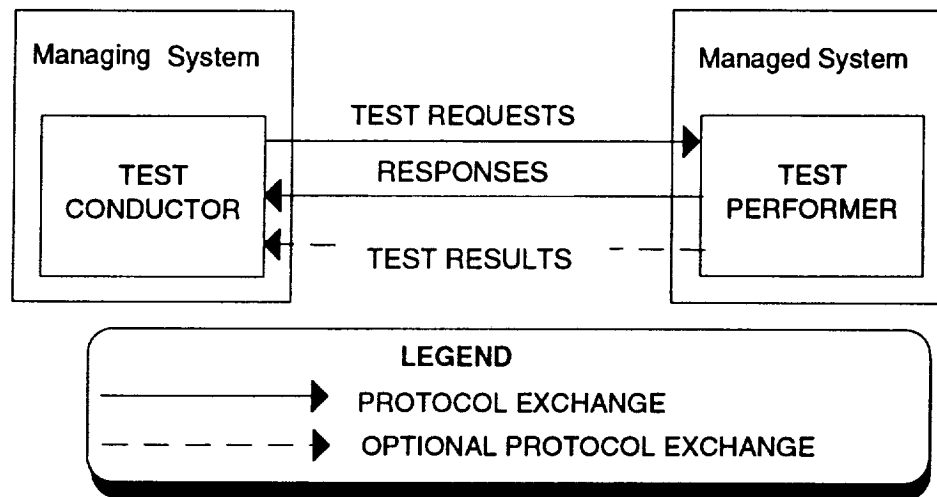


Figure 20. The Test Function Model

The testing objects⁷² (TOs) initialized and enabled by the TARR capability generate test requests that are required for the control and monitoring of tests and for the emission of notifications pertaining to tests. TOs exist only for the duration of the test. An individual test may have any number of TOs. A TO is uniquely identified and named by a testing object ID. The testing object ID may be assigned either by the test conductor or the test performer. A TO may include a schedule mechanism or refer to a schedule function that allows the initiation and termination time of the test to be controlled.

⁷¹ Test action request receiver (TARR): The test action request receiver is a term used to identify the ability of a managed object to act upon a test request. (For example, the SMI is a TARR since it can receive test commands and invoke test sequences.)

⁷² Testing object (TO): A testing object is a managed object that exists only for the duration of a test invocation and which has attributes, behaviours, and event notifications that pertain to that instance of test. It issues the specific test, measures the test responses, and generates the resulting test event notifications.

The managed objects under test⁷³ (MOTs) provide the management views of the subject of tests are identified in test requests. Each test must involve one or more MOT. The execution of the test relies upon mechanisms provided by the TOs and the MOTs.

Tests may be defined as being synchronous⁷⁴ or asynchronous⁷⁵. A synchronous test is one in which the final results of the test are returned in the response to the test initiation operation (command). Figure 21 illustrates the synchronous test. The asynchronous test is one in which the final results of the test are to be made available by some further management operation or via an event notification issued by the TO. Figure 22 illustrates the asynchronous test.

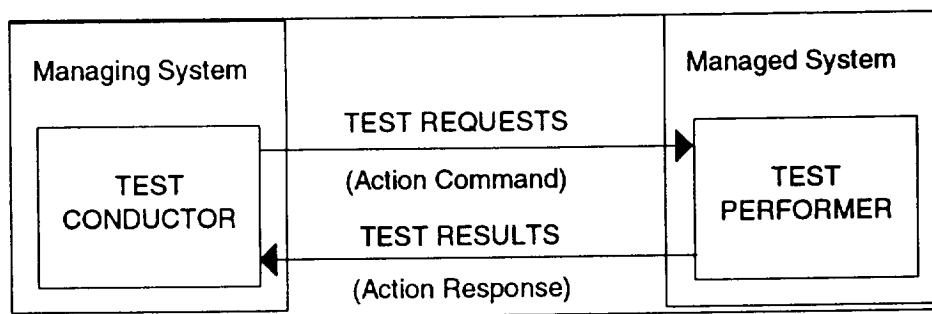


Figure 21. The Synchronous Test

⁷³ Managed object under test (MOT): The managed object under test is the managed object that represents a management view of the resource or resources whose functions are the subject of the test.

⁷⁴ Synchronous test: A synchronous test is a test invocation for which any confirmation to the test request implies termination of the test invocation.

⁷⁵ Asynchronous test: An asynchronous test is a test invocation for which the successful confirmation to the test request does not imply termination of the test invocation.

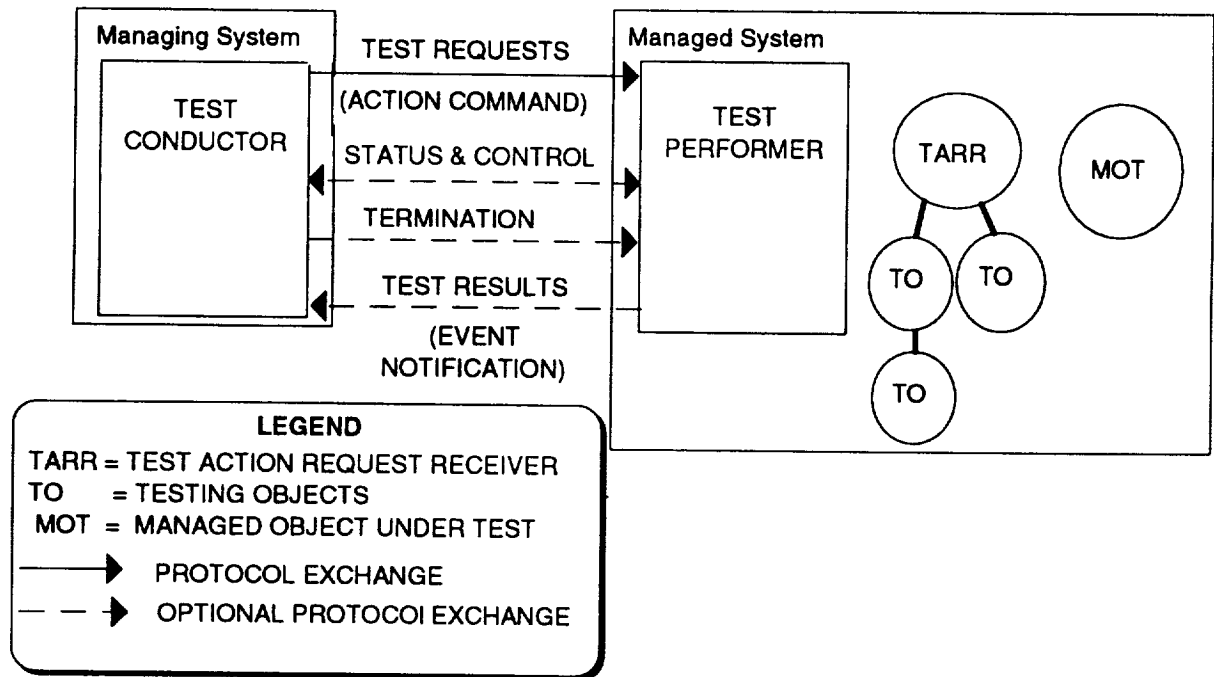


Figure 22 . The Asynchronous Test

The committee draft ISO/IEC 10164-12 specifies that the asynchronous tests are modeled using separate managed objects to represent the TARR and the TO. In the asynchronous environment, the TO is initialized and enabled by the TARR, and it is contained within the TARR as a result of the a test action request. A TO exists only for the asynchronous test invocations. The TO holds test status, and intermediate and unreported test results associated with the invocation⁷⁶. Intermediate test results are those that have not yet been conveyed to the test conductor. Requests to suspend, resume, or terminate are directed to the object with TARR capability (the ISE). The affected TOs are identified using either a list of the TOs names or a test session⁷⁷ identifier.

The committee draft ISO/IEC 10164-12 specifies the content of the test request. It includes the following information carried in the parameter of the request.

⁷⁶ Test invocation: A test invocation is a specific instance of test, from the time of initiation to termination.

⁷⁷ Test session: A test session is a set of test invocations.

- The type of the TARR input information
- The parameter specific to the type of TARR input information
- The TOs required
- The information pertaining to initial attribute values for the TOs
- The identity of the MOTs

The committee draft ISO/IEC 10164-12 specifies the test state attributes of TOs as the following seven distinct states and rules related to the test management function:

- The testing states are as follows:
 - **Idle off-duty:** The test is enabled but idle waiting for the scheduled start or for the next schedule iteration or has passed the scheduled stop time.
 - **Idle on-duty:** The test has not yet been initiated due to busy conditions. For example, the test may be a process in a queue and awaiting execution. In this transient state the test is started.
 - **Initiating:** That period during which the test is being set up. A test may be in this state for a significant length of time if it is required to wait for the managed object to be configured.
 - **Suspended:** A test may be suspended by the suspend request. In this state, it is not executing, but its read attributes are readable, and its write attributes can be commanded. For example, while a test is suspended test result attributes may be accessed or scheduling attributes may be changed. The resume request continues the testing.
 - **Testing:** The testing state reflects the active phase of the test during which the testing algorithms and measurements are taking place.
 - **Terminating:** The test environment transitions to a dormant condition. This includes the activities that are necessary to restore the test resources or MOTs to their pre-test condition.

- **Disabled:** The test is disabled when it becomes totally inoperable due to failure conditions. This may be temporary or lead to abnormal test termination⁷⁸.
- The following rules apply to the testing states:
 - Some classes of testing objects exhibit a subset of these states. For example, a test which requires no particular testing environment might not need any set up time and may not exhibit the initiating state nor the terminating state. Likewise, a test which does not support scheduling would not exhibit the idle off-duty state.
 - All test support test-result attributes, the test-result attributes may become and remain accessible at any time except during the initiating state.

The committee draft specifies the services of test suspension, resumption, and termination. The committee draft specifies that the operation of these services must follow a consistent set of rules related to the saving of information and orderly transitions depending on the testing states.

The committee draft specifies the reporting of the test results for the synchronous and asynchronous test. Test result notifications contain the test session identifier, if it was used. The test result notification indicates that it is sending no more reports by including the test outcome attribute in a report. The test outcome attribute takes the value PASS, FAIL, or INCONCLUSIVE. The interpretation of the test parameter depends on the type of tests requested. If the test outcome attribute indicates FAIL, then the notification may contain parameters indicating the nature of the problem and proposed repair action (see event report function).

The committee draft specifies that complex and compound tests can be conducted using the test management function. A test conductor may request a number of test performers to initiate tests concurrently (by using a schedule). In this way, a test conductor may collect information from several different systems concerning the set of circumstances under investigation. Also, a test performer may involve one or more systems in performing the test requested by the test conductor. In this case, an application process in the system containing the test performer acts as a subsidiary test conductor and requests test performers in other

⁷⁸ Abnormal test termination: Abnormal test termination is a statement made with respect to a test invocation when the test is prematurely terminated.

systems to initiate the test. (For example, the SSCC test conductor could request the ISE to conduct a station-wide test involving two or more system's test. The ISE acts as the test conductor for the SSCC and involves the test conductors in the systems, elements, or payloads.) Such tests are known as complex tests.

The committee draft specifies the use of the following attributes:

- A **test section identifier** attribute, which indicates a set of test invocations. If a test session is used, then the test conductor (i.e., the ISE, SSCC, or POIC) assigns the test section identifier.
- A **MOTs** attribute, which identifies the managed object instance(s) that represents the resource being used or to be tested.
- The **test state** attribute, which identifies the current state of a test invocation.
- The **test outcome** attribute, which provides a standard view of test results: **PASS**, **FAIL**, or **INCONCLUSIVE**.
- The **associated objects** attribute, which identifies the managed object instance(s) that represent another resource involved in the test.
- The **time-out period** attribute, which sets the maximum amount of time that a test may be in the testing state.
- The **test object identifier** attribute, which indicates the testing object.

The committee draft specifies the following parameters to be included either in test management services or in test management event notification:

- The mandatory **test request ID** parameter may include input information needed by the test performer to initiate a test, but which is not required to be held in the TO attributes. The parameter identifies the syntax of the corresponding test request information parameter.
- The optional **test request information** parameter has its syntax identified by the **test request ID**. It conveys input information needed by the test performer to execute a test, but which is not required to be held in any TO attributes. The test request information is registered using a label of ACTION-INFO context type,

associated with the test action registered⁷⁹ in the ACTION section of a TARR package template (as defined in clause 10.4 of international standard ISO/IEC 10165-4.)

- The **test object list** parameter specifies the testing objects that are to be created as a result of an asynchronous test request. For each testing object in the list, it includes the TO class and may include a TO instance name as well as TO attribute initialization information.
- The **optional test object result list** parameter conveys the names of the testing objects created as a result of an asynchronous test request.
- The **async test result ID** parameter identifies the syntax of a corresponding asynchronous test result information parameter.
- The **async test result information** parameter holds the information for a particular test result notification. The async test result information is registered using a parameter label of EVENT-INFO context type, associated with the test result notification listed in the NOTIFICATION section of TO package template (as defined in clause 10.4 of ISO 10165-4).
- The **synch test result ID** parameter identifies the syntax of a corresponding synchronous test result information parameter.
- The **synch test result information** parameter holds the information for a particular test result notification. The synch test result information parameter is registered using a parameter table of ACTION-REPLY context type. The synch test result information parameter is associated with the synchronous test request action listed in the ACTION section of TARR package template.

⁷⁹ **Registration:** Registration is the process of defining and putting under configuration management control the definitions and transfer syntax of attributes, objects, notifications (and TOLs), actions (commands), packages, behaviours, and relationships. The ISO/IEC standard 10165 specifies how to complete the template (forms) necessary to "freeze" a managed object. Registration involves some authority (e.g., NASA configuration management change boards) to publish and duplicate the information for others to use. The registration of an object makes it an object with defined interfaces and properties that others can communicate with and obtain its services.

- The test failure parameter is included in the failure confirmation notification to a test request as information describing the failure. This parameter takes on the following specified values:
 - Test request not supported
 - No such MOT
 - Invalid MOT classs
 - MOT not available
 - Argument error
 - Duplicate TO instance
 - Unable to create TO
 - Subsidiary test failure
 - Reason not specified

J.2 Notifications of the Test Management Function

The committee draft defines one notification of the test management function:

- The test result notification

The committee draft maps the parameters of the notification to the CMIS event report service. The committee draft in clause 11 maps the test result notification parameters to the CMIS parameters.

Also the committee draft specifies the use of services to modify the operation of the test management function. In particular, the operations specified use the following standard test function services.

- The test request async service
- The test request sync service
- The test suspend-resume service
- The test termination service

The ISO/IEC CD 10164-12, clause 11, provides the mapping of the parameters of these services to the CMIS M-ACTION parameters.

J.3 Attributes and Objects for Objects and Attributes for Test Management

The attributes, parameters, and objects for representing the test management functions will be provided by the detail design of the DMS, ISE and the *Freedom's* objects, systems, elements, and payloads. The DMS STSV should have testing objects and attributes for a test management function to meet the needs of the SSFP. Examples of how DMS could provide the testing objects and attributes for the test management function are provided in the ISO/IEC CD 10164-12. The designs of the ISE and DMS do not have to comply with ISO/IEC CD 10164-12 but the capability of DMS will require the functionality of the standard.

The attributes for the testing objects and attributes for the test management function are defined and explained by the ISO/IEC CD 10164-12.

The ISO/IEC CD 10164-12 specifies the following attributes for the test management function:

- TestSessionID
- TestState
- TestOutcome
- MOTS
- AssociatedObjects
- TimeoutPeriod

J.4 Protocol and Abstract Syntax Definitions of Objects and Attributes for Test Management

The ISO/IEC 10164-12 committee draft defines the ASN.1 value notations for all the objects, attributes, notifications, actions (commands), and behaviours needed by the objects and attributes for the test management function. These abstract syntax definitions will move to ISO/IEC 10165-2 when ISO/IEC 10164-12 becomes an international standard.

APPENDIX K

THE SCHEDULING FUNCTION

This section of the document describes the objects and attributes for the scheduling⁸⁰ function that may be used by the application processes in the Space Station *Freedom* Program. The scheduling function is proposed to meet the needs of the SSFP Tier 1 to separate schedules from the application processes that need schedules. External schedulers were described as desirable in the logging (journalizing) function, the summarization function, and in the event forwarding function. Scheduling of these functions was allowed by building the scheduling attributes into the applications of those functions. However, it would be efficient to have scheduling objects (schedule packages) that many applications could share and which would allow some degree of synchronization of activities. A DMS STSVs could supply these scheduling objects. (Perhaps a few would meet most scheduling requirements: one would meet the requirements for logging, another would meet the requirements for TOL generation, and another would meet the requirements for testing).

The scheduling function needs to provide for the following user needs:

- The ability to schedule a number of activities within multiple managed objects by a single scheduler.
- The ability to specify the time duration that the schedule is active.
- The ability to schedule the control interval of operation of an activity within a managed object, the start time should be defined as the actual time, and the stop time should be definable using either actual time or an allowable offset from the start time.
- The ability to provide interval scheduling⁸¹ and periodic scheduling⁸².

⁸⁰ Scheduling function: The method of controlling the timing of the performance of a scheduled activity within an object are represented by another managed object.

⁸¹ Interval scheduling: Interval scheduling is a type of scheduling that controls a number of intervals of operation of activities within specified managed object instances.

⁸² Periodic scheduling: This is a type of scheduling that controls the repetitive triggering of activities within specified managed object instances.

- Provide a defined number of intervals together with the start and stop times of each interval with the specified period.
- Provide a configuration schedule that repeats over a given period.

Standard scheduling objects and attributes that provide these basic needs would form a part of a systematic and flexible scheduling structure. The following sections include descriptions of the scheduling function proposed as a standard by the performance management committee working draft, ISO/IEC 10164-s. These sections include scheduling managed objects, their attributes, and their event notifications that meet the above needs. Section 5 of this document includes tradeoffs, issues, and risks associated with supplying a DMS STSV with this design of scheduling objects, attributes, and event notifications.

K.1 The Model of the Scheduling Function

The SSFP objects, systems, elements, and payloads are to be defined in accordance with appendix D, the Flight Software Data and Object Standard of the DMS ACD, (NASA, 1991 [SSP30261]). This data standard refers to an applicable document, the SMI ISO/IEC 10165. SMI part 2 will contain the ISO attributes and objects used for the scheduling function when the scheduling function working draft (WD) ISO/IEC 10164-s, becomes an international standard. The model of the scheduling function's objects is not stable. It has to pass many international revisions before it becomes stable; however, it does provide a design for a scheduling function, and it could be used as a set of level C requirements for a DMS STSV. Currently the committee WD, ISO/IEC 10164-s, defines syntax for the scheduling attributes, objects and notifications. This working draft describes how scheduling objects and their attributes and their event notifications would work together to meet the Tier 1 scheduling needs. It also consistently defines terms that comply with the *Basic Reference Model* (ISO, 1984 [7498-1]), the *Open System Management Framework* (ISO, 1989 [7498-4]), the CMIS (ISO, 1990 [9595]), the *Open System Management Overview* (ISO, 1991 [10040]), and the other parts of ISO/IEC 10164.

The committee working draft ISO/IEC 10164-s specifies the terminology concerned with scheduling. The model for scheduling objects is illustrated in figure 23. The model consists of the scheduler object⁸³ (SO) that supplies schedules for scheduled managed objects⁸⁴ (SMOs). The scheduler object has the behaviour that supplies the scheduling function to the scheduled managed objects. A scheduler object is controlled by the managing system and sends event notifications to the managing system.

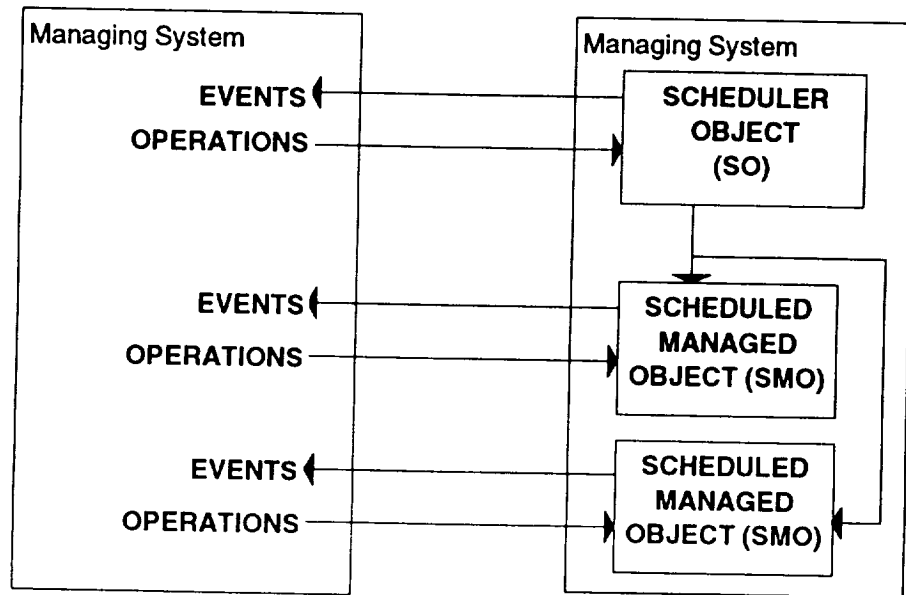


Figure 23. The Scheduling Function Model

The committee working draft specifies two capabilities for the scheduler object: internal and external scheduling. The internal scheduling capability is when the scheduler object is contained within the SMOs. The external scheduling capability is when a SO is not contained within the SMO. Multiple SMOs may be triggered by the same SO. External scheduling eliminates the need to replicate and co-ordinate scheduling data across the SMO instances. Figure 23 illustrates the external scheduling capability.

⁸³ Scheduler object (SO): A scheduler object is the managed object that defines the type and values of the schedule to be applied to activities within the SMOs.

⁸⁴ Scheduled managed object (SMO): The scheduled managed object is the managed object whose activities are to be scheduled.

The committee working draft specifies a pointer within the SMO to identify the SO for its behaviour. The committee working draft specifies the inclusion of attributes in the SO to point to the SMOs.

The committee working draft specifies two behaviours for both the SO and SMO. These two behaviours are interval scheduling behaviour and periodic scheduling behaviour. To provide the behaviour, a set of conditional packages are defined. These conditional packages are as follows:

- Daily scheduling
- Weekly scheduling
- Multiple daily scheduling
- Multiple weekly scheduling
- Monthly scheduling
- Periodic scheduling
- Activation duration package
- Duration package

The committee working draft specifies new attributes to control the scheduling function. These attributes are as listed:

- The **expiry behaviour** attribute specifies the action to be taken by the SO when the stop time has been exceeded.
- The **external schedulers** attribute specifies the SO instances together with the activities that are controlled by them within the SMOs.
- The **mode** attribute defines the mode of synchronization of a periodic schedulers triggering periods upon suspension and resumption to the SMO. If the value is **TRUE**, it implies that the triggering period will be synchronized to the presuspended triggering points when the operation of the SMO has been suspended and resumed. If the value is **FALSE**, it implies that when the operation of the SMO has been suspended it triggers on resumption and synchronizes the period to the resumption triggering point.
- The **period attribute** defines a time period in terms of days, hours, minutes, or seconds.

- The **scheduled activities** attribute specifies the SMO instances and the activities within those instances that are controlled by the SO.
- The **scheduler ID** attribute is the distinguished attribute for naming the instance of a SO.
- The **sequence of days** is a structured attribute that defines a sequence of intervals of day. The interval of a day component within the sequence of days attribute type defines a list of time intervals (interval-start and interval-end times of day). If the values of the attribute are not specified in the schedule initialization and enablement, then its value defaults to a single interval encompassing the entire 24-hour period of a day.
- The **sequence of months** is a structured attribute defining a sequence of month masks. Each month mask is a set of mask components. Each mask component specifies a set of time intervals on a 24-hour time-of-day clock pertaining to selected days of the month. The attribute consists of two components: days of month and intervals of day. If not present, they default to every day of the month and the entire 24-hour period of a day, respectively.
- The **sequence of weeks** is a structured attribute defining a sequence of week masks. Each week mask is a set of mask components. Each mask component specifies a set of time intervals on a 24-hour time-of-day clock pertaining to selected days of the week. The attribute consists of two components: days of week and intervals of day. If not present, they default to every day of the week and the entire 24-hour period of a day, respectively.
- The **start time** attribute defines the start day and time.
- The **stop time** attribute defines the stop day and time.

K.2 Notifications of the Scheduling Function

The committee working draft does not define any notification of the scheduling function other than those with the management of the SO. The object management notifications specified are as follows:

- AttributeValueChange
- ObjectCreation

- ObjectDeletion

Also the committee working draft leaves the use of services to modify the operation of the scheduling function to be determined. However, if the scheduling service is to be useful, it should include the following typical ISO/IEC 10164 standard services.

- The suspend-resume service
- The termination service

K.3 Attributes and Objects for Objects and Attributes for the Scheduling Function

The attributes, parameters, and objects for representing the scheduling functions will be provided by the detail design of the DMS, ISE and the *Freedom's* objects, systems, elements, and payloads. The DMS STSV should have scheduling objects and attributes for a scheduling function to meet the needs of the SSFP. Examples of how DMS could provide the scheduling objects and attributes for the scheduling function are provided in the ISO/IEC WD 10164-s. The designs of the ISE and DMS do not have to comply with ISO/IEC WD 10164-s, but the space station scheduling objects and attributes for the scheduling function will require equivalent functions.

The ISO/IEC WD 10164-s specifies the following attributes for the scheduling function:

- ExpiryBehaviour
- ExternalSchedulers
- Mode
- Period
- Scheduled activities
- ScheduledID
- SequenceOfDays
- SequenceOfMonths
- SequenceOfWeeks
- StartTime
- StopTime

K.4 Protocol and Abstract Syntax Definitions of Objects and Attributes for the Scheduling Function

The ISO/IEC 10164-s committee working draft defines the ASN.1 value notations for all the objects, attributes, actions (commands), and behaviours needed by the objects and attributes for the scheduling function. These abstract syntax definitions will move to ISO/IEC 10165-2 when ISO/IEC 10164-s becomes an international standard.

GLOSSARY

ACC	access control certificate
ACD	Architectural Control Document
ACI	access control information
ADF	access control decision function
AEF	access control enforcement function
ANSI	American National Standards Institute
APM	Columbus Attached Pressurized Module
ARD	Action Reader Directory
ASN.1	Abstract Syntax Notation One
BIT	built-in test
BITE	built in test equipment
C&W	caution and warning system
C&T	Communications & Tracking
CCSDS	Consultative Committee for Space Data Systems
CEI	contract end item
CD	committee draft
CHeCS	Man Systems/Crew Health Care System
CMIS	Common Management Information Services
COTS	commercial-off-the-shelf
CSCI	computer software configuration item
DID	data item description
DMS	Data Management System
ECLSS	Environment Control Life Support System
EM	Element Manager
EPS	Electrical Power System
ETCS	External Thermal Control System
EVA	Extra Vehicular Activities
EVAS	Extra Vehicular Activity System
FSSR	Flight Software System Requirements
FDIR	failure detection, isolation, and recovery
GN&C/P	Guidance Navigation & Control/Propulsion

ICD	interface control document
ID	identification
ILE	Intermediate Language Executor
IODB	Input/Output Data Base
IP	international partners
IS	international standards
ISE	Integrated Station Executive
ISO	International Organization for Standardization
ITCS	Internal Thermal Control System
JSC	Johnson Space Center
JEM	Japanese Experiment Module
LAN	local area network
MDM	Multiplexer/Demultiplexer
MDSSC	McDonnell Douglas Space Systems Company
MODB	Master Objects Data Base
MSC	Mobile Servicing Center
MSC	Management Service Control
MTE	Mobile Transporter Element
NASA	National Aeronautics and Space Administration
NOS	Network Operating System
OSI	Open System Interconnection
OSTP	on-board short term plan
ORU	Orbit Replaceable Units
POIC	Payload Operations and Integration
RJ	Rotary Joint
RODB	Runtime Object Database
SDMC	Station Docking Mast Controller
SDP	Standard Data Processor
SM	Station Manager
SMI	Structure of Management Information
SMM	System and Mission Mangement
SMO	scheduled managed object
SO	scheduler object

SPCP	stored program command procedure
SSCC	Space Station Control Center
SSFP	Space Station <i>Freedom</i> Program
STSV	standard services
TARR	test action request receiver
TOL	telemetry object list
ULC/PAMS	Unpressurized Logistics Carrier/ Propulsion Module Attachment
USE	User Support Environment

Distribution List

INTERNAL

H010

S. W. Gouse
L. W. Thomas
R. E. Smylie

C022

Technical Report Center (2)

D045

R. C. LaBonte (A155)

W155

Records Resources (2)

MITRE-Washington Technical Centers

W153

Artificial Intelligence Technical Center (W410)

W115

Economic Analysis Technical Center (W960)

W153

Modeling and Simulation Technical Center (W410)

W031

Network Technical Center (W420)

G147

Security Technical Center (Z269)

W096

Signal Processing Technical Center (Z407)

W150

Software Technical Center (Z289)

G149

Systems Engineering Technical Center (Z532)

F085

C. Doug Morris (W389) (3)

MITRE-Bedford Technical Centers G110

Artificial Intelligence Technical Center (K312)

D90

Cost Analysis Technical Center (G103)

G100

Network Technical Center (K318)

D50

Reliability & Maintainability (H114)

D80

Sensor Technical Center (N210)

D70

Software Technical Center (A350)

D80

VLSI/VHSIC Technical Center (E095)

D115

Claude E. LaBarre

H120

J. S. Brown
C. T. Curtin
A. H. Ghovanlou
J. C. Heberlig
D. G. Rea
Houston Technical Library (2)

H121

W. M. Evanco

H122

S. D. Richmond (Z379)
E. L. Tilton (Z385)

H123

J. K. Richardson
R. W. Zears

H124

L. P. Seidman

H125

R. M. Jackson
E. L. Berger (3)

H126

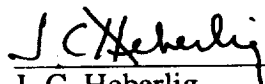
C. R. Somerlock (Z386)

EXTERNAL

NASA

Technical Library (3) - JM2

APPROVED FOR PROJECT
DISTRIBUTION:

 11/9/92

J. C. Heberlig
Associate Technical Director, H120

Bergu (MITRE)
NASA SCIENTIFIC AND TECHNICAL DOCUMENT AVAILABILITY AUTHORIZATION (DAA)

To be initiated by the responsible NASA Project Officer, Technical Monitor, or other appropriate NASA official for all presentations, reports, papers, and proceedings that contain scientific and technical information. Explanations are on the back of this form and are presented in greater detail in NHB 2200.2, "NASA Scientific and Technical Information Handbook."

☒ Original
☐ Modified

(Facility Use Only)
Control No. _____
Date _____

I. DOCUMENT/PROJECT IDENTIFICATION (Information contained on report documentation page should not be repeated except title, date and contract number)
Title: Integrated Station Executive Requirements & System Design Approach

Author(s): Eugene L. Berger C. Doug Morris

Originating NASA Organization: NASA/JSC, KG2

Performing Organization (if different) NAS9-18057

Contract/Grant/Interagency/Project Number(s) WP 92W0000302/MTR92W0000247; NASA CR-185708

Document Number(s) _____

(For presentations or externally published documents, enter appropriate information on the intended publication such as name, place, and date of conference, periodical or journal title, or book title and publisher: _____)

These documents must be routed to NASA Headquarters, International Affairs Division for approval. (See Section VII)

II. AVAILABILITY CATEGORY

Check the appropriate category(ies):
Security Classification: ☐ Secret ☐ Secret RD ☐ Confidential ☐ Confidential RD ☒ Unclassified

Export Controlled Document - Documents marked in this block must be routed to NASA Headquarters International Affairs Division for approval.

☐ ITAR ☐ EAR
NASA Restricted Distribution Document ☐ Special Conditions-See Section III

☐ FEDD ☐ Limited Distribution
Document disclosing an invention

☐ Documents marked in this block must be withheld from release until six months have elapsed after submission of this form, unless a different release date is established by the appropriate counsel. (See Section IX).

Publicly Available Document

☒ Publicly available documents must be unclassified and may not be export-controlled or restricted distribution documents.
☐ Copyrighted ☐ Not copyrighted

III. SPECIAL CONDITIONS

Check one or more of the applicable boxes in each of (a) and (b) as the basis for special restricted distribution if the "Special Conditions" box under NASA Restricted Distribution Document in Section II is checked. Guidelines are provided on reverse side of form.

a. This document contains:
☐ Foreign government information ☐ Commercial product test or evaluation results

☐ Other-Specify _____

b. Check one of the following limitations as appropriate:
☐ NASA contractors and U.S. Government agencies only
☐ U.S. Government agencies and U.S. Government agency contractors only
☐ NASA personnel and NASA contractors only ☐ NASA personnel only

☐ Preliminary information

☐ Information subject to special contract provision

☐ U.S. Government agencies only

IV. BLANKET RELEASE (OPTIONAL)

All documents issued under the following contract/grant/project number _____ Date _____ is:
The blanket release authorization granted _____ is:
☐ Rescinded - Future documents must have individual availability authorizations.

☐ Modified - Limitations for all documents processed in the STI system under the blanket release should be changed to conform to blocks as checked in Section II.

V. PROJECT OFFICER/TECHNICAL MONITOR

J. T. Chapman III
Typed Name of Project Officer/Technical Monitor

KG2
Office Code

J. T. Chapman III
Signature

1/4/93
Date

VI. PROGRAM OFFICE REVIEW

Richard H. Knapp
Typed Name of Program Office Representative

☒ Approved
☐ Not Approved
M-8
Program Office and Code

Signature

1/27/93
Date

VII. INTERNATIONAL AFFAIRS DIVISION REVIEW

☐ Open, domestic conference presentation approved.
☐ Foreign publication/presentation approved.
☐ Export controlled limitation is approved.

☐ Export controlled limitation is not applicable.
☐ The following Export controlled limitation (ITAR/EAR) is assigned to this document: _____

International Affairs Div. Representative

Title

Date

VIII. EXPIRATION OF REVIEW TIME

The document is being released in accordance with the availability category and limitation checked in Section II since no objection was received from the Program Office within 20 days of submission, as specified by NHB 2200.2, and approval by the International Affairs Division is not required.

Name & Title

Office Code

Date

IX. DOCUMENTS DISCLOSING AN INVENTION

a. This document may be released on _____ Date _____

Installation Patent or Intellectual Property Counsel

NASA STI Facility

Date

b. The document was processed on _____ Date _____

in accordance with Sections II and III as applicable.

X. DISPOSITION

Completed forms should be forwarded to the NASA Scientific and Technical Information Facility, P.O. Box 8757, B.W.I. Airport, Maryland 21240, with either (check box):
☐ Printed or reproducible copy of document enclosed
☐ Abstract or Report Documentation Page enclosed. The issuing or sponsoring NASA installation should provide a copy of the document, when complete, to the NASA Scientific and Technical Information Facility at the above listed address.

INSTRUCTIONS FOR COMPLETING THE NASA SCIENTIFIC AND TECHNICAL DOCUMENT AVAILABILITY AUTHORIZATION (DAA) FORM

The DAA Form, or its equivalent, is used to prescribe the availability and distribution of all NASA-generated and NASA-funded documents containing scientific and technical information. Either a suitable description (title, abstract, etc.) of the document or a completed copy must accompany this form. This form requires an appropriate Program Office review and approval, and in some cases, an International Affairs Division review and approval. The Center Representative for Document Availability Authorization should forward the completed DAA to the NASA Scientific and Technical Information Facility on completion. Specific guidelines for each Section follow:

- I. **Document/Project Identification:** Provide the information requested. If the document is classified, indicate security classification of the title and abstract. (Classified information must not be entered on this Form). Include RTOP numbers under the Contract number entry. Provide information on presentations or externally published documents as applicable. Documents intended for domestic presentation or publication must be approved in accordance with NASA STI Handbook (NHB 2200.2) while documents intended for presentation must also be coordinated with the appropriate NASA installation or NASA Headquarters, Scientific and Technical Information Branch in accordance with NHB 2200.2. Note that information on the Report Documentation Page (if attached) is not to be entered here except for title, document date, and contract number.
- II. **Availability Category:** Check the appropriate category or categories.
 - Security Classification.** Enter the applicable security classification for the document. Documents, if classified, will be available to all appropriately cleared personnel having a "need to know".
 - Export Controlled Document.** If the document meets the provision of NHB 2200.2 Paragraph 5(g), the appropriate restriction must be checked, either International Traffic in Arms Regulations (ITAR) or Export Administration Regulations (EAR). This form must then be routed to the International Affairs Division for completion of Section VII. This category cannot be used with NASA Restricted Distribution Documents.
 - NASA Restricted Distribution Document.** If the document meets the provisions of NHB 2200.2, Paragraph 5(b), then the appropriate restriction must be checked, either "For Early Domestic Dissemination" (FEDD), or "Limited Distribution." If other special conditions apply to document availability, check the "Special Conditions" box and use Section III to determine the basis for such determination and the special handling required. This category cannot be used with Export Controlled Documents.
 - Document Disclosing an Invention.** This box must be checked when documents contain information which discloses an invention. When this box is checked, an additional appropriate availability category must be checked. Authorization for use of this category must be provided by Installation Patent Counsel in Section IX.
 - Publicly Available Document.** Check this box if the document is to be made available to the general public without restrictions. If this box is checked please indicate whether the document is copyrighted or not according to paragraph 203.2a in NHB 2200.2.
- III. **Special Conditions:** These boxes are checked only when the box designated "Special Conditions" in Section II has been checked. Both (a) and (b) are to be completed.
 - This subsection (a) describes the information content:
 - Foreign government information.** Information provided by foreign governments under special agreements or the results of jointly sponsored research and development with agreed to limitations.
 - Commercial product test or evaluation results.** Information resulting from the testing and/or evaluation of commercial products or processes that may unduly affect them if published.
- IV. **Blanket Release:** This optional Section is to be completed whenever subsequent documents produced under the contract, grant or project are to be given the same distribution and/or availability as described in Section II. More than one contract number or RTOP Number can be entered. This Section may also be used to rescind or modify an earlier Blanket Release. All blanket releases must be approved by the Program Office (or its designee) and the International Affairs Division (if applicable), and concurred in by the Office of Management.
- V. **Project Officer/Technical Monitor:** The Project Officer or Technical Monitor should sign and date the form. The office code and typed name should be entered. The date signed should reflect the submission date to the program office whose approval will be entered in Section VI.
- VI. **Program Office Review:** This Section is to be completed by the duly authorized official representing the Program Office. Any delegation from NASA Headquarters to Field Installations in accordance with NHB 2200.2 should be entered here.
- VII. **International Affairs Division Review:** This Section is to be completed by the authorized representative of the International Affairs Division for all documents intended to be Export Controlled, for foreign publications or presentations, and for open domestic conference presentations.
- VIII. **Expiration of Review Time:** NHB 2200.2 provides twenty days for Program Office and International Affairs Division review. If no review has been received within twenty days, the Technical Monitor or Project Officer may release the document as marked in Section II. This release cannot be used for Export Controlled Documents, Conference presentations, or foreign publications.
- IX. **Document Disclosing an Invention:** In part a of this Section, the Installation Patent Counsel or the Intellectual Property Counsel may release a document in a time frame other than six months by entering the date and signing the alternate release. The NASA Scientific and Technical Information Facility will process and distribute these documents after six months in accordance with Sections II and III unless otherwise notified.
- X. **Disposition:** This form, when completed, is to be sent to the NASA Scientific and Technical Information Facility, P.O. Box 8757, B.W.I. Airport, Maryland 21240. When available, a printed or reproducible copy of the document should be sent with the form; otherwise, an abstract or Report Documentation Page should be sent. Forms that contain availability categories that have been disapproved by the Program Office or the International Affairs Division may be returned to the initiating NASA Technical Monitor or Project Office for resubmission.

Preliminary information. Preliminary or incomplete results, studies or recommendations for which wider distribution would be premature.

Special contract provisions. Information developed under NASA contracts that contain provisions providing limited rights to the data generated.

Other. Information that should be restricted for other reasons. The specific reason must be entered after "Other".

This subsection (b) on limitations refers to the user groups authorized to obtain the document. The special limitations apply both to the initial distribution of the documents and the handling of requests for the documents. The limitations will appear on and apply to reproduced copies of the document. Documents limited to NASA personnel should not be made available to on-site contractors. If approval of the issuing office is checked, the NASA Scientific and Technical Information Facility will provide only bibliographic processing and no initial distribution; the Facility will refer all document requests to the issuing office.